Default Log Rotation Behavior

By default, MongoDB uses the *--logRotate rename* behavior. With `rename`, `mongod` or `mongos` renames the current log file by appending a UTC timestamp to the filename, opens a new log file, closes the old log file, and sends all new log entries to the new log file.

Step 1: Start a `mongod` instance.

```
mongod -v --logpath /var/log/mongodb/server1.log
```

You can also explicitly specify *logRotate --rename*.

Step 2: List the log files In a separate terminal, list the matching files:

```
ls /var/log/mongodb/server1.log*
```

The results should include one log file, `server1.log`.

Step 3: Rotate the log file. Rotate the log file by issuing the `logRotate` command from the `admin` database in a `mongo` shell:

```
use admin
db.runCommand( { logRotate : 1 } )
```

Step 4: View the new log files List the new log files to view the newly-created log:

```
ls /var/log/mongodb/server1.log*
```

There should be two log files listed: `server1.log`, which is the log file that `mongod` or `mongos` made when it reopened the log file, and `server1.log.<timestamp>`, the renamed original log file.

Rotating log files does not modify the "old" rotated log files. When you rotate a log, you rename the `server1.log` file to include the timestamp, and a new, empty `server1.log` file receives all new log input.

Log Rotation with `--logRotate reopen`

New in version 3.0.0.

Log rotation with *--logRotate reopen* closes and opens the log file following the typical Linux/Unix log rotate behavior.

Step 1: Start a `mongod` instance, specifying the `reopen` *--logRotate* behavior.

```
mongod -v --logpath /var/log/mongodb/server1.log --logRotate reopen --logappend
```

You must use the *--logappend* option with *--logRotate reopen*.

Step 2: List the log files In a separate terminal, list the matching files:

```
ls /var/log/mongodb/server1.log*
```

The results should include one log file, `server1.log`.