

Restart tests, located in [tests/restarting](#), test FDBs ability to correctly function after an upgrade or downgrade. Tests come in pairs, with `*-1.txt` to be run first, and `*-2.txt` to be run second. Folders matching the expression `from_x.x.x` test upgrading the binary by one version (i.e. testing an upgrade from 6.3.5 to 7.0.0), while folders matching `to_x.x.x` test downgrading the binary by one version.

Reproducing a restart failure is different from other simulation tests. Specifically, to reproduce the failure, we need to run two commands, one with the old binary and one with the new binary. Before running a restart test, make sure you have two binaries of the correct FDB version available. You can [download old binaries](#) or just build the version you need.

Note: Joshua [docker image](#) relies on the above download urls for restarting tests.

First, run the first restart test with the correct binary. For upgrade tests (`from_x.x.x`), use the older binary. For downgrade tests (`to_x.x.x`), use the newer binary. This example will look at an upgrade test.

```
$ ../build6.2/bin/fdbserver -r simulation -f ../foundationdb/tests/restarting/from_6.2.0/SnapTestSimpleRestart-1.txt --seed 523887594 --buggify on
```

Next, from the same folder, run a simulation of the **second** restart test with the **other** binary. For upgrade tests, use the newer binary. For downgrade tests, use the older binary. **Additionally, make sure to pass the `--restarting` flag and to use the seed + 1 and buggify value.**

```
$ ./bin/fdbserver -r simulation -f ../foundationdb/tests/restarting/from_6.2.0/SnapTestSimpleRestart-2.txt --seed 523887595 --buggify on --restarting
```

Debugging Joshua

A Joshua failure may produce output like:

```
<Test TestUID="7e8c7574-74fe-487f-8af3-05421d646384" RandomSeed="535104944" SourceVersion="dd5481fb60ee16576.0.3" TestFile="/app/deploy/ensembles/20190123-161607-jingyu_zhou-bcdbea9b64b020d3/tests/restarting/CycleTestRestart-1.txt" SimElapsedTime="59.8124" RealElapsedTime="1.9464" />
<Test TestUID="7e8c7574-74fe-487f-8af3-05421d646384" RandomSeed="535104945" SourceVersion="f7ed11ccd60596876.0.3" TestFile="/app/deploy/ensembles/20190123-161607-jingyu_zhou-bcdbea9b64b020d3/tests/restarting/CycleTestRestart-2.txt" SimElapsedTime="2110.19" RealElapsedTime="74.412" />
<DisableConnectionFailures_Tester Severity="30" LogGroup="default" />
<TestFailure Severity="40" Error="timed_out" ErrorDescription="Operation timed out" ErrorCode="1004" Reason="e fdbserver.debug -p -C -f -i 0x1479f46 0x14786b7 0xad9537 0xad9ab7 0x45dfc9 0x7b13d4 0x530ae0 0x141a2d3 0x141a2d3" />
<CodeCoverage File="fdbserver/SimulatedCluster.actor.cpp" Line="1293" />
<CodeCoverage File="fdbserver/DiskQueue.actor.cpp" Line="616" />
...(elided)
```

To determine the version of the old binary, look for an `OldBinary` field in the `<Test>` trace. In the example above, `OldBinary="fdbserver-6.0.3"`.

If you are unable to reproduce the failure, check if the compiler you used is the same as Joshua uses (GCC 8). Keep in mind that different Linux versions can also affect determinism, so you may want to reproduce using the [development docker image](#).