```
1  ---------- MODULE datasync ----------
2  EXTENDS Naturals, TLC, Sequences, FiniteSets, Integers, ExternalSeqRecordParser2  导入标准模块或者自定义模块
3  VARIABLE leader
4  VARIABLE follower
5  VARIABLE followerSendEpochToLeader
6  VARIABLE leaderReceivedEpochFromFollower
7  VARIABLE leaderCalculaterNewEpoch
8  VARIABLE followerReceivedNewEpochFromLeader
9  VARIABLE leaderSyncDataZxid
10 VARIABLE followerSyncDataZxid
11 VARIABLE offset
12 vars ≜ ⟨leader, follower, followerSendEpochToLeader, leaderReceivedEpochFromFollower, leaderCalculaterNe
13  Trace ≜ ExSeqRcdParser2("D:\\00001code\\runtimemodel\\zookeeper_environment\\datasync.log")

15   leader选举结束开启数据同步过程，数据同步过程分为如下阶段
16     1.follower发送epoch给leader
17     2.leader接受follower发送的epoch
18     3.leader重新计算epoch，为newEpoch，并发送newepoch给follower
19     4.follower接受newEpoch，并发送ack消息给leader
20     5.leader接收到newepoch，开始数据同步，发送相应的Zxid
21 Init ≜
22         ∧ leader = 0
23         ∧ follower = 0
24         ∧ followerSendEpochToLeader = 0
25         ∧ leaderReceivedEpochFromFollower = 0
26         ∧ leaderCalculaterNewEpoch = 0
27         ∧ followerReceivedNewEpochFromLeader = 0
28         ∧ leaderSyncDataZxid = 0
29         ∧ followerSyncDataZxid = 0
30         ∧ offset = 1
```

```
Trace ≜ ExSeqRcdParser2("./datasync.log")

Rule ≜ ∧ offset ≤ Len(Trace)
       ∧ offset' = offset + 1
       ∧ leader' = Trace[offset].leader
       ∧ follower' = Trace[offset].follower
       ∧ followerSendEpochToLeader' = Trace[offset].followerSendEpochToLeader
       ∧ leaderReceivedEpochFromFollower' = Trace[offset].leaderReceivedEpochFromFollower
       ∧ leaderCalculaterNewEpoch' = Trace[offset].leaderCalculaterNewEpoch
       ∧ followerReceivedNewEpochFromLeader' = Trace[offset].followerReceivedNewEpochFrom
       ∧ leaderSyncDataZxid' = Trace[offset].leaderSyncDataZxid
       ∧ followerSyncDataZxid' = Trace[offset].followerSyncDataZxid
       Rule1
       ∧ Assert(leaderCalculaterNewEpoch' > followerSendEpochToLeader',
                "Newepoch > epoch")
       Rule2
       ∧ Assert(leaderSyncDataZxid' = followerSyncDataZxid',
                "lastzxid equales")
       Rule3
       ∧ Assert(followerSendEpochToLeader' = leaderReceivedEpochFromFollower',
                "Epoch value equals")
       Rule4
       ∧ Assert(leaderCalculaterNewEpoch' = followerReceivedNewEpochFromLeader',
                "NewEpoch value equals")

get ≜
    ∧ offset ≤ Len(Trace)
    ∧ offset' = offset + 1
    ∧ leader' = Trace[offset].leader
    ∧ follower' = Trace[offset].follower
    ∧ followerSendEpochToLeader' = Trace[offset].followerSendEpochToLeader
    ∧ leaderReceivedEpochFromFollower' = Trace[offset].leaderReceivedEpochFromFollower
    ∧ leaderCalculaterNewEpoch' = Trace[offset].leaderCalculaterNewEpoch
    ∧ followerReceivedNewEpochFromLeader' = Trace[offset].followerReceivedNewEpochFromLea
```

$\quad\quad \land\ leaderSyncDataZxid' = Trace[offset].leaderSyncDataZxid$

$\quad\quad \land\ followerSyncDataZxid' = Trace[offset].followerSyncDataZxid$

$term \triangleq \land\ offset > Len(Trace)$

$\quad\quad \land\ \text{UNCHANGED}\ vars$

$Next \triangleq \lor\ get$

$\quad\quad \lor\ term$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$EpochConsistency \triangleq followerSendEpochToLeader = leaderReceivedEpochFromFollower$

$NewEpochConsistency \triangleq leaderCalculaterNewEpoch = followerReceivedNewEpochFromLeader$

$SyncDataZxidConsistency \triangleq leaderSyncDataZxid = followerSyncDataZxid$

---

* Modification History

* Last modified *Thu Mar* 17 16:13:46 *CST* 2022 by 10222803

* Created *Mon Feb* 21 13:54:48 *CST* 2022 by 10222803