

# CIP-94 On-chain DAO Vote for Chain Parameters

Conflux [CIP-94\(On-chain DAO Vote for Chain Parameters\)](#) ParamsControl, DAO Conflux

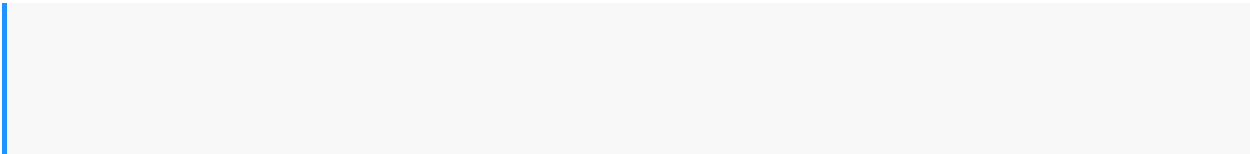


CFX

- 1. Staking C votePower
  - 1. deposit
  - 2. voteLock
- 2. ParamsControl castVote

ParamsControl — CIP-94

ParamsControl



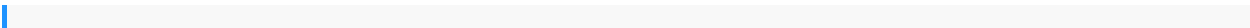
round/vote\_round 2 \* 60 \* 60 \* 24 \* 60 blockNumber Conflux ParamsControl currentRound()

CIP-94 “PoW” “PoS”

- “ ”
- “ ” 2
- “ ” 1/2

x x-1 old new n\_unchange, n\_increase, n\_decrease “ ” “ ”

$$new = old * 2^{\frac{n_{increase} - n_{decrease}}{n_{unchange} + n_{increase} + n_{decrease}}}$$



- “ ” “ ”
- `[1/2 * old, 2 * old]` “ ” “ ”
- “1”

`ParamsControl` `castVote(uint64 vote_round, Vote[] vote_data)` `vote_round` `vote_data` “Vote” `Vote`

```
struct Vote {
    // topic_index      0    PoW      1    PoS
    uint16 topic_index;
    // votes[0], votes[1], votes[2]    “ ” “ ” “ ”
    // Vote votes      votePower
    uint256[3] votes;
}
```

`ParamsControl` `castVote(uint64 vote_round, Vote[] vote_data)`

```
(0, [100, 0, 100])    topic_index = 0    votes[0] = 100, votes[2] = 100 )
(0, [100, 0, 100]), (1, [0, 0, 200]) (0, [0, 0, 0]) , (0, [0, 0, 0]), (1, [0, 0,
200]) topic_index
```

CIP-105 [CIP-105](#) PoS 5%

# js

1. `Staking` `C` `votePower`
  1. `deposit`
  2. `voteLock`
2. `ParamsControl` `castVote`

## Staking

## Staking

```
//      Staking
const { Conflux } = require('js-conflux-sdk');

async function main() {
  //
  const PRIVATE_KEY = '0xxxxxxx'; //
  const cfx = new Conflux({
    url: 'https://portal-test.confluxrpc.com',
    networkId: 1,
    logger: console,
  });
  const account = cfx.wallet.addPrivateKey(PRIVATE_KEY);

  const staking_contract = cfx.InternalContract('Staking');
  //  deposit(uint amount)      Drip,      1CFX  1e18 Drip
  const stake_amount = BigInt(1e18);
  await staking_contract.deposit(stake_amount).sendTransaction({
    from: account
  }).executed();
  console.log("Stake finished")

  //  voteLock(uint amount, uint unlock_block_number)      Drip
  //      1CFX
  const lock_amount = stake_amount;
  //      CFX      1      1 Drip      1
  //      1
  // Conflux      Conflux
  //      1.01      1      blockNumber + 1.01
  const current_block_number = (await cfx.getStatus())['blockNumber']
  const unlock_block_number = current_block_number + 2 * 60 * 60 * 24 * 365 * 1.01
  await staking_contract.voteLock(lock_amount, unlock_block_number).sendTransaction({
    from: account
  }).executed();
  console.log("Lock finished")

  const votePower = await staking_contract.getVotePower(account.address, "")
  console.log(`vote power: ${votePower}`) // vote power: 1000000000000000000
```

```
}
```

```
main()
```

# ParamsControl

CIP-94

```
const { Conflux } = require('js-conflux-sdk');

async function main() {
  //
  const PRIVATE_KEY = '0xxxxxxxxx';
  const cfx = new Conflux({
    url: 'https://portal-test.confluxrpc.com',
    networkId: 1,
    // logger: console,
  });
  const account = cfx.wallet.addPrivateKey(PRIVATE_KEY);

  const control_contract = cfx.InternalContract('ParamsControl');
  //
  const current_round = await control_contract.currentRound()
  const base_amount = BigInt(1e17);
  //
  let vote0 = [0, [base_amount, base_amount, base_amount]]
  let vote1 = [1, [base_amount, base_amount, 0]]
  await control_contract.castVote(current_round, [vote0, vote1]).sendTransaction({
    from: account
  }).executed();
  console.log("Cast vote finished")

  //
  const vote_result = await control_contract.readVote(account.address)
  console.log(vote_result)

  //
  const all_votes = await control_contract.totalVotes(current_round)
  console.log(all_votes)
}
```

```
main()
```

---

Revision #20

Created 15 July 2022 10:32:05 by Darwin

Updated 27 September 2022 08:44:15 by Darwin