



Advisor Role on the Peerplays Blockchain

Peerplays utilizes Graphene’s governance structures to reach on-chain consensus about business management decisions and blockchain parameters. Consensus needs to be reached by the Advisors Board (also known as the Committee). Members of the Board (the advisors) are voted in by the holders of the governance token PPY.

The Peerplays blockchain has a set of parameters available that are subject to Committee approval. PPY token holders can have an opinion on their preferred set of parameters and thereby support an existing Advisor or alternatively become an Advisor themselves.

The Peerplays Advisor Committee consists of several active Advisors (‘active’ meaning, as with block producers, that they have received enough votes to currently be able to execute the duties of a Peerplays Advisor). The number of active Advisors as well as the support threshold to become active is controlled by PPY token holders.

Some of the Peerplays parameters include:

- fee structure:** fees that have to be paid by customers for individual transactions
- blockchain parameters interval:** i.e. block interval, max size of block/transaction
- witness parameters:** i.e. maximum amount of witnesses (block producers)
- advisor parameters:** i.e. maximum amount of advisor members

A full list of the parameters present in the Peerplays blockchain is listed in the document [Peerplays-Bulletin-05June2017.pdf](#).

Additionally to defining the parameters any active Witness can propose a protocol or business upgrade (i.e. hard fork) which can be approved by shareholders.

1. CREATING A NEW ADVISOR

Every existing account on the PeerPlays Blockchain can create one Advisor/Committee member. We can create a new Advisor with:

```
>>> create_committee_member account "url" true
```

2. HOW TO PROPOSE ADVISOR ACTIONS

The cli-wallet knows two calls that are handy when proposing a change to the blockchain parameters and fees:

- a. `propose_fee_change`: This call allows to propose a change in one or multiple fees. It works as follows (example):

```
propose_fee_change proposer "exp" {"transfer":{"fee":10,"price_per_kbyte":1}} False
proposer: Proposer account: any account
exp: Expiration time in the format: "2017-11-28T11:54:59"
```

- b. `propose_parameter_change`: This allows to propose the change of a single blockchain parameter, for instance:

```
propose_parameter_change proposer "exp" {"block_interval":2} False
proposer: Proposer account: any account
exp: Expiration time in the format: "2017-11-28T11:54:59"
```



After creation of a proposal, an object id of the form 1.10.x will be created that identifies this particular proposal. The list of proposals can be found from the advisory board account that has the name `committee-account`.

3. HOW TO APPROVE/DISAPPROVE AN ADVISOR PROPOSAL

3.1 APPROVING PROPOSAL

Now we need to convince the other Advisors to approve the proposal. We can do so on the blockchain by asking them for approval with

```
>>> approve_proposal <fee-paying-account> <proposal-id>
{"active_approvals_to_add" : ["<MY-ADVISOR>"]} true
```

where `<proposal-id>` takes the form `1.10.xxx` and identifies the actual proposal to approve.

3.2 REMOVING APPROVAL

A previous approval can also be removed if the proposal is not yet expired, executed or within the preview period. This is done by:

```
>>> approve_proposal <fee-paying-account> <proposal-id>
{"active_approvals_to_remove" : ["<MY-ADVISOR>"]} true
```

Note that we now use `active_approvals_to_remove` instead of `active_approvals_to_add`.

4. HOW ADVISORS PROPOSE A CHANGE IN FEE

4.1 IDENTIFY THE FEE TO CHANGE

All fees are accessible from the object `2.0.0` and has the following form:

```
{...,
  parameters: {
    current_fees: {
      parameters: [
        [operation-id, {fee-details}],
        [operation-id, {fee-details}],
        ...
        [operation-id, {fee-details}],
      ],
      scale: scale
    }
  }
}
```

In order to identify the actual fee for an operation (such as the asset creation operation), we first need to identify the operation id for `transfer_operation` which can be done by looking at the blockchain or by using

```
get_prototype_operation account_create_operation
```

in the cli-wallet. The first element in the array is the operation id (here: 5).

The corresponding fee takes the form:

```
{
  "account_create_operation" : {
    "basic_fee" : 500000,
    "premium_fee" : 200000000,
  }
}
```



```
    "price_per_kbyte": 100000}
}
```

The *actual* fee that needs to be paid for a basic account is

$$500000 / scale$$

The *scale* parameter can be obtained from the parameters above.

4.2 CREATE A PROPOSAL

Let's assume we want to propose a new fee for the account creation operation. We want 5 PPY as basic fee and want premium names to cost 2000 PPY. Additionally, a price per kbyte for the account creation transaction can be defined. We get:

```
{
  "account_create_operation" : {
    "basic_fee"      : 500000,
    "premium_fee"   : 200000000,
    "price_per_kbyte": 100000}
}
```

We propose the fee change for account <advisor> with:

```
>>> propose_fee_change <advisor> "2015-10-14T15:29:00"
{"account_create_operation" : {"basic_fee": 500000, "premium_fee":
200000000, "price_per_kbyte": 100000}} false
```

4.3 APPROVE PROPOSAL

Now we need to convince the other committee members to approve. We can do so on the blockchain by asking them for approval with

```
>>> approve_proposal <committee_member>
"1.10.1" {"active_approvals_to_add" : ["<MY-ADVISOR>"]} true
```

where 1.10.1 is the id of the proposal in question.

=====

ADDENDUM

Many of the operations described in this document can be performed more simply with the python-peerplays library (with the exception of creation of the actual proposal).

For example, a proposal can be approved with

```
peerplays approveproposal --account <approver> <proposal-id>
```