Proton / XPR based Staking and FREEOS Economy

@Author: Mohammad Shahid Siddiqui
@Email: mssiddiqui.nz@gmail.com

Introduction

This document describes the precise commands to execute the actions of the smart contract freeos (currently in testnet it is freeos333333, but when live it will be only freeos unless stated otherwise).

The smart contract freeos has main actions that a user has to execute in order to contribute into the FREEOS economy.
There is another smart contract named freeosconfig which contains all the configurations that freeos consumes.

The user follows the process:

1. Creates an account along with a wallet comprising of public:private key pairs for active and owner permissions.
2. Registers the account with the Proton Net (Mainnet or Testnet as per the deployment). He calls reguser action to register to the freeos.
3. Stakes the XPRs as described in the staking requirement, by calling the transfer action of eosio.token smart contract. It depends upon iteration number as well as the total users staked before him.
4. Now every week (or iteration) he claims, by calling the claim action of freeos contract.

The claim action mints the FREEOS currency and as per design some minted FREEOS goes to the user (liquid and vested) while some goes to the freeosdivide contract. The distribution depends upon the iteration number.

Create New Proton Accounts

Run the script created by Shahid. Ensure that the account type is one of the types: ‘e’, ‘d’, and ‘v’.
$ bash CreateAccounts.sh

Initialise Account and Password in Shell

acc=<account>

```
psw=<wallet password>
```

Unlock Wallet

```
proton wallet unlock -n proton_${acc} --password ${psw}
```

[Make sure that wallet is present in current dir: proton_${acc}]

E.g. proton wallet unlock -n proton_eddieeagle --password
PW5J3cVcn8BQqWPox6KrGR2F5EK9cDjvcKeTxC25my4axnwtVKDTN

List Wallets and check if a wallet is unlocked

```
proton wallet list
```

Check in local wallet folder (e.g. ~/eosio-wallet/) what accounts have wallets:

```
ls  -ltr pr*wallet| cut -d"." -f1|cut -d "_" -f2
```

Get the staking requirement for the current iteration

```
proton get table freeos333333 freeos333333 stakes
```

Iterations- View & Update

Update the iteration table

#Bear in mind that times are UTC time zone (currently 13 hours behind NZT)
The Arguments are: [Iteration Num, Start Time, End Time, Claim Amount, Tokens Required in Iteration]

Iteration 1 - Update Time and Current Rate

```
proton push action freeosconfig iterupsert '[1, "2021-02-28 19:00:00", "2021-03-01 09:59:59", 100, 0]' -p freeosconfig
```

Iteration 2 - Update Time and Current Rate

```
proton push action freeosconfig iterupsert '[2, "2021-03-01 10:00:00", "2021-03-01 10:59:59", 125,
0]' -p freeosconfig
```

Iteration 3 - Update Time and Current Rate

```
proton push action freeosconfig iterupsert '[3, "2021-03-01 11:00:00", "2021-03-01 12:59:59", 150,
0]' -p freeosconfig
```

Iteration 4 - Update Time and Current Rate

```
proton push action freeosconfig iterupsert '[4, "2021-03-01 19:00:00", "2021-03-02 09:59:59", 175,
0]' -p freeosconfig@active
proton push action freeosconfig currentrate '[0.0150]' -p freeosconfig@active
```

Iteration 5 - Update Time and Current Rate

```
proton push action freeosconfig iterupsert '[5, "2021-03-02 10:00:00", "2021-03-03 00:59:59", 200,
275]' -p freeosconfig@active
```

```
11:15:31 02/03/2021 ~/eosio-wallet$ proton push action freeosconfig iterupsert '[5, "2021-03-02 10:00:00",
"2021-03-03 00:59:59", 200, 275]' -p freeosconfig@active
executed transaction: 679565e90df4e02c6b8b38173563f6015f2adf7478a3adb9974aa6b9545aea68  144 bytes
200 us
#  freeosconfig <= freeosconfig::iterupsert     {"iteration_number":5,"start":"2021-03-02 10:00:00","end":"2021-
03-03 00:59:59","claim_amount":200,"...
>> iteration: 5 start: 2021-03-02 10:00:00, end: 2021-03-03 00:59:59, claim amount: 200, tokens_required:
275 added to the iterations table
```

Change Current Rate for Iteration 5

```
proton push action freeosconfig currentrate '[0.0003]' -p freeosconfig@active
```

```
11:15:43 02/03/2021 ~/eosio-wallet$ proton push action freeosconfig currentrate '[0.0003]' -p
freeosconfig@active
executed transaction: 8fd3c523bc1b1b23f726ccdebe39935df722b6fba4d8e6dedc00322af608ae25  104 bytes
181 us
#  freeosconfig <= freeosconfig::currentrate    {"price":"0.00030000000000000"}
>> Current exchange rate set to 3.0000000000000000e-04
warning: transaction executed locally, but may not be confirmed by the network yet         ]
11:16:20 02/03/2021 ~/eosio-wallet$
```

Get the current Exchange Rate

```
proton get table freeosconfig freeosconfig exchangerate
```

🔍Refer this sheet for more iterations and corresponding rates:

https://docs.google.com/spreadsheets/d/1XCA-3SBGl5TGIfOu3NJI9l7x1wb3tmI0ccbqUzX-YEA/edit?ts=6024b267#gid=1380083826

Iteration #8

```
proton push action freeosconfig iterupsert '[8,"2021-03-05 07:00:00", "2021-03-06 07:59:59",275,613]' -p freeosconfig@active
proton push action freeosconfig currentrate '[0.0184]' -p freeosconfig@active
```

View Iteration Details

```
proton push action freeosconfig getiter '[1]' -p freeosconfig@active
proton push action freeosconfig getiter '[2]' -p freeosconfig@active
```

#Look into the all Iterations

```
proton get table freeosconfig freeosconfig iterations
```

Sample output would be:

```
proton get table freeosconfig freeosconfig iterations
{
  "rows": [{
      "iteration_number": 1,
      "start": 1614538800,
      "start_date": "2021-02-28 19:00:00",
      "end": 1614589200,
      "end_date": "2021-03-01 09:00:00",
      "claim_amount": 100,
      "tokens_required": 0
    },{
…
   },{
      "iteration_number": 10,
      "start": 1615201200,
      "start_date": "2021-03-08 11:00:00",
      "end": 1615287599,
      "end_date": "2021-03-09 10:59:59",
      "claim_amount": 325,
      "tokens_required": 900
    }
  ],
  "more": false,
  "next_key": ""
}
```

Get Target Price and Exchange information

```
#Get the FREEOS target price and current price aka exchange rate-
proton get table freeosconfig freeosconfig exchangerate
```

#RegUser

```
proton push action freeos333333 reguser "[\"${acc}\"]" -p ${acc}@active
```

#GetUser

```
proton push action freeos333333 getuser "[\"${acc}\",\"e\"]" -p ${acc}@active
```

#Check Users and Claims table in FreeOs contract

```
proton get table freeos333333 ${acc} users
proton get table freeos333333 ${acc} claims
```

#Unvest

```
proton push action freeos333333 unvest "[\"${acc}\"]" -p ${acc}@active
```

#Check the accounts status

```
proton get currency balance eosio.token ${acc} XPR
proton get currency balance freeos333333  ${acc} FREEOS
proton get currency balance eosio.token freeos333333 XPR
proton get currency balance freeos333333 freeos333333 FREEOS
proton get currency balance freeos333333 freedao33333 FREEOS
proton get currency balance freeos333333 freeosdivide FREEOS
```

#Stake (Transfer)

```
proton push action eosio.token transfer "[\"${acc}\", \"freeos333333\",\"0.0000 XPR\", \"freeos stake\"]"  -p $acc@active
```

#Claim

```
proton push action freeos333333 claim "[\"${acc}\"]" -p $acc@active
```

Sample would be similar to it:

```
10:26:28 08/03/2021 ~/eosio-wallet$ proton push action freeos333333 claim "[\"${acc}\"]" -p $acc@active
executed transaction: 8723931cbcd2965f3175d21b879125dd9844d7b3bf2825ea581f5fbd186c8d85  104 bytes  377 us
#  freeos333333 <= freeos333333::claim          {"user":"lushalusha"}
>> lushalusha claimed 30.0000 FREEOS and vested 270.0000 FREEOS for iteration 9
#  freeos333333 <= freeos333333::issue          {"to":"freeos333333","quantity":"6330.0000 FREEOS","memo":"claim by lushalusha"}
#  freeos333333 <= freeos333333::transfer       {"from":"freeos333333","to":"lushalusha","quantity":"30.0000 FREEOS","memo":"claim by lushalusha"}
#  freeos333333 <= freeos333333::transfer       {"from":"freeos333333","to":"freeosdivide","quantity":"6300.0000 FREEOS","memo":"claim by lushalusha...
#    lushalusha <= freeos333333::transfer       {"from":"freeos333333","to":"lushalusha","quantity":"30.0000 FREEOS","memo":"claim by lushalusha"}
#  freeosdivide <= freeos333333::transfer       {"from":"freeos333333","to":"freeosdivide","quantity":"6300.0000 FREEOS","memo":"claim by lushalusha...
warning: transaction executed locally, but may not be confirmed by the network yet         ]
```

If claim is tried in the same iteration again, an error will be thrown:

```
10:27:03 08/03/2021 ~/eosio-wallet$ proton push action freeos333333 claim "[\"${acc}\"]" -p $acc@active
Error 3050003: eosio_assert_message assertion failure
Error Details:
assertion failure with message: user is not eligible to claim in this iteration
pending console output: user lushalusha has already claimed in claim period 9
```

Check the FREEOS currency after the claim action

```
proton get currency balance freeos333333  ${acc} FREEOS
```

918.0000 FREEOS

#Unstake

```
proton push action freeos333333 unstake "[\"${acc}\"]" -p $acc@active
```

Unlock Admin Wallets

Freeosconfig:

```
pswconfig=<Enter password of freeosconfig wallet>

proton wallet unlock -n proton_freeosconfig --password ${pswconfig}
---
```

Freeos333333:

```
pswfreeos=<Enter password of freeos333333 wallet>
proton wallet unlock -n proton_freeos333333 --password ${pswfreeos}
----
```

Getting Staked Users Information

Get users who have staked

```
proton get scope freeos333333 -t users --limit 1000| grep scope
```

Then look at each user record individually. e.g.

```
proton get table freeos333333 eddieeagle users
```

Get claims count and staked count (unlock wallet freeos333333)

```
proton push action freeos333333 getcounts '[]' -p freeos333333@active
>> users registered: 6, claim events: 2 unvest percent: 0
```

Get XPRs for Accounts

#E.g. To transfer XPR to account eddieeagle:

```
proton push action eosio.token transfer '["freeos333333", "eddieeagle", "10.0000 XPR", "loan"]' -p freeos333333

proton push action eosio.token transfer "[\"freeos333333\", \"${acc}\", "10.0000 XPR", "loan"]' -p freeos333333
```

#Get XPR to account: Open Browser and get the Coins
https://monitor.testnet.protonchain.com/#faucet

Transfer FREEOS to any account

To transfer (say) 100 FREEOS from the freeos333333 account to (say) billbeaumont, it would be:

```
proton push action freeos333333 transfer '["freeos333333", "billbeaumont", "100.0000 FREEOS", "loan"]' -p freeos333333@active
```

Version of Contract

```
proton push action freeos333333 version '[]' -p ${acc}@active
```

```
>> Version = 0.329 - it is currently iteration 8
```