

BeKind Stake Pools Online (BKIND)



DISCLAIMER: This document is supplied by BeKind Stake Pools Online. Usage of the content of this document is at your own risk. Under no circumstances will BeKind Stake Pools Online be held responsible or liable in any way for any claims, damages, losses, expenses, costs or liabilities whatsoever (including, without limitation, any direct or indirect damages for loss of profits, business interruption or loss of information) resulting or arising directly or indirectly from your use of or inability to use this document, or from your reliance on the information and material in this document, even if BeKind Stake Pools Online has been advised of the possibility of such damages in advance.

Table of Contents

<i>Introduction.....</i>	<i>3</i>
<i>Retrieve required information from the blockchain.....</i>	<i>4</i>
<i>Calculate the transaction fee for the transaction.....</i>	<i>5</i>
<i>Building the transaction.....</i>	<i>7</i>
<i>Signing the transaction.....</i>	<i>8</i>
<i>Submitting the transaction.....</i>	<i>9</i>

Introduction

To save time and to limit transaction costs, it is possible to send lovelaces (ada) to multiple addresses in a single transaction. This document describes the commands to be issued with `cardano-cli` to construct and execute the (multi-receiver) transaction.

Note: In some commands the files containing addresses are used. In other commands you will need to use the actual addresses.

Retrieve required information from the blockchain

To be able to use `cardano-cli`, make sure you have a fully synced Cardano node running on the machine you are executing the `cardano-cli` commands from.

1. First we need to get the blockchain settings. These can be retrieved by issuing the following command:

```
cardano-cli query protocol-parameters --mainnet --out-file protocol.json
```

2. For the transaction we need to pay a transaction fee. To get the amount of lovelaces (1000000 lovelaces = 1 ada) available in the payment (sender) address, issue the following command:

```
cardano-cli query utxo --address $(cat payment address) --mainnet
```

payment address: enter the full path to the file containing the payment address with which you want to pay for the transaction fee

3. The `ttl` sets the time on which the transaction should be completed before it gets cancelled. The `ttl` (slot number) is a blockchain time value. We need to add some extra time to the current `ttl`, because a transaction can only be done at a certain time in the future. The current `ttl` can be determined by executing the command below:

```
cardano-cli query tip --mainnet
```

Calculate the transaction fee for the transaction

For each transaction on the Cardano Blockchain one needs to pay some transaction fee. To calculate the fee, the transaction to be executed has to be constructed. In the example given below, one address (wallet) is used for sending lovelaces (1 ada = 1000000 lovelaces) and for paying the transaction fee. The lovelaces will be sent to three different addresses (wallets).

1. Execute the following command to build the transaction for calculating the fee:

```
cardano-cli transaction build-raw \  
--tx-in TxHash of the sender#TxIx of sender \  
--tx-out sender address+0 \  
--tx-out receiver address 1+0 \  
--tx-out receiver address 2+0 \  
--tx-out receiver address 3+0 \  
--invalid-hereafter ttl plus some extra time \  
--fee 0 \  
--out-file tx.raw
```

TxHash of the sender#TxIx of sender

Replace with the transaction TxHash and TxIx information holding the lovelaces of the sender (this is the transaction with the sender address holding the lovelaces). This information is retrieved in previous paragraph step 2.

sender address

Replace with the address of the sender (this is the address from which the fee of the transaction will be paid and from which the lovelaces will be sent to the three receiver addresses). This is the address (payment address) used in previous paragraph step 2.

receiver address 1

Replace with the address of receiver 1.

receiver address 2

Replace with the address of receiver 2.

receiver address 3

Replace with the address of receiver 3.

ttl plus some extra time

Replace with the ttl value retrieved in previous paragraph step 3 + 2000. Adding 2000 to mentioned value is good practise and should give enough time to prepare and execute the commands in the next paragraphs.

The command above with the replaced values can now be executed.

2. In this step the transaction fee is going to be calculated based upon the transaction build in the previous step. Execute the command below to calculate the fee:

```
cardano-cli transaction calculate-min-fee \  
--tx-body-file tx.raw \  
--mainnet \  
--protocol-params-file protocol.json \  
--tx-in-count value tx-in \  
--tx-out-count value tx-out \  
--witness-count value witness-count \  
--byron-witness-count 0
```

value tx-in

The transaction consists of 1 tx-in (see step 1), so replace with 1.

value tx-out

The transaction consists of 4 tx-out (see step 1), so replace with 4.

value witness-count

Payments will be done from 1 address, so we need to sign the transaction with 1 secret key (the secret key of the payment address), so replace with 1.

The command above with the replaced values can now be executed. The fee is returned.

Building the transaction

All information needed to build the transaction is now available. In this example the sender address has 20000000 lovelaces (20 ada) in a transaction. To each receiver 5000000 lovelaces (5 ada) will be sent. In this example the fee as retrieved in the previous paragraph is 188953 lovelaces. The ttl retrieved + 2000 is 69323784. Before the transaction can be build, the amount of lovelaces that need to be sent back to the sender address has to be calculated.

1. To calculate which amount needs to be sent back to the sender address:

Amount of lovelaces in the transaction of the sender - the sum of lovelaces to be sent to the recievers - transaction fee

$$20000000 - 15000000 - 188953 = 4811047$$

2. The transaction has to be as follows:

```
cardano-cli transaction build-raw \  
--tx-in TxHash of the sender#TxIx of sender \  
--tx-out sender address+4811047 \  
--tx-out receiver address 1+5000000 \  
--tx-out receiver address 2+5000000 \  
--tx-out receiver address 3+5000000 \  
--invalid-hereafter 69323784\  
--fee 188953 \  
--out-file tx.raw
```

TxHash of the sender#TxIx of sender

Replace with the transaction TxHash and TxIx information holding the lovelaces of the sender (this is the transaction with the sender address holding the lovelaces). This information is retrieved earlier.

sender address

Replace with the address of the sender (this is the address from which the fee of the transaction will be paid and from which the lovelaces will be sent to the three receiver addresses). This is the sender address (payment address) used earlier.

receiver address 1

Replace with the address of receiver 1.

receiver address 2

Replace with the address of receiver 2.

receiver address 3

Replace with the address of receiver 3.

The command above to build the transaction can now be executed.

Signing the transaction

The transaction build in the previous paragraph can now be signed.

Sign the transaction with the follow command:

```
cardano-cli transaction sign \  
--tx-body-file tx.raw \  
--signing-key-file path to file holding the secret key of payment address \  
--mainnet \  
--out-file tx.signed
```

path to file holding the secret key of payment address

Replace with the full path to the file holding the secret key of the payment address.

Proceed to the next paragraph to submit the transaction to the blockchain.

Submitting the transaction

Submit the transaction to the blockchain by executing the command below:

```
cardano-cli transaction submit \  
--tx-file tx.signed \  
--mainnet
```

The lovelaces (ada) will now be send to the receivers.

In case of any questions, please do not hesitate to contact us on Telegram.