



adax

CARDANO TOKEN MINTING AND BURNING

SUMMARIZED BY ADAX.PRO

DEVELOPER GUIDE FOR CARDANO TOKEN CREATION:

Following is the reference guide link to create the tokens on the cardano blockchain

<https://docs.cardano.org/en/latest/native-tokens/getting-started-with-native-tokens.html#example-minting-a-new-native-token>

NODE STARTUP:

01 Start cardano relay node (Test Network)

02 Use this configuration file while starting the node:

```
wget https://hydra.iohk.io/build/5102327/download/1/launchpad-config.json
wget https://hydra.iohk.io/build/5102327/download/1/launchpad-byron-genesis.json
wget https://hydra.iohk.io/build/5102327/download/1/launchpad-shelley-genesis.json
wget https://hydra.iohk.io/build/5102327/download/1/launchpad-topology.json
wget https://hydra.iohk.io/build/6498473/download/1/testnet-alonzo-genesis.json
```

03 Configuration parameters to start the node:

--topology	"path to the testnet-topology.json file"
--socket	"path to the node.socket file"
--config	"path to the testnet-config.json"
--database-path	"path to the database directory"
--port	"3001"

Note: Path of "node.socket" file is required. Upon node startup, it will be created automatically by the relay node.

04 Set the environment variable "CARDANO_NODE_SOCKET_PATH":

```
export CARDANO_NODE_SOCKET_PATH=~/.PWD/Cardano/RelayNode/node.socket
```

Note: Wait few hours for relay node synchronization with Cardano test network.

Testnet Explorer:

```
https://explorer.cardano-testnet.iohkdev.io/en.html
```

Testnet Faucet:

```
https://developers.cardano.org/en/testnets/cardano/tools/faucet/
```

PEE-REQUISITES:

The "cardano-cli" command-line tool will be provided by the relay node. It supports many wallet functionalities that include address generation, transaction creation, signing, broadcasting, etc.

To mint, the token/ asset on the Cardano, issuer account, and script policy is required. Therefore, "cardano-cli" can be used for this.

01 Create a keypair:

```
cardano-cli address key-gen \  
--verification-key-file policy.vkey \  
--signing-key-file policy.skey
```

Note "policy.vkey" and "policy.vkey" are the names of a public key and private key.

Using the above-described command, two files for a key pair are created under the names “policy.skey” and “policy.vkey”. The contents in these files can be displayed using the “cat” command.

```
#cat policy.skey
{
  "type": "PaymentSigningKeyShelley_ed25519",
  "description": "Payment Signing Key",
  "cborHex":
  "*****"
}
```

```
#cat policy.vkey
{
  "type": "PaymentSigningKeyShelley_ed25519",
  "description": "Payment Signing Key",
  "cborHex":
  "5820985f84712a23a1bb47cecf13c5ce8e49d633105a94f5c826ae597f4b9bbc3be1"
}
```

02 Derive the payment address from using the “cardano-cli address build” command

```
./cardano-cli address build \
--payment-verification-key-file policy.vkey \
--out-file pay.addr \
--testnet-magic 1097911063
```

Note the “pay.addr” is the output file where payment address is stored

Print the payment address

```
# cat pay.addr

addr_test1vqfnzcx8amctaw2zqmnhrt2ns80n2pqnkjdzcdymakj5rnct09mxh
```

Query this payment address in the testnet explorer:

```
https://explorer.cardano-testnet.iohkdev.io/en/address?address=addr_test1vqfnzcx8amctaw2zqmnhrt2ns80n2pqnkjdzcdymakj5rnct09mxh
```

Query this address using the command “cardano-cli query utxo”

```
# ./cardano-cli query utxo \  
--address addr_test1vqfnzcx8amctaw2zqgmhrt2ns80n2pqnkjdzcdymakj5rnct09mxh  
--testnet-magic 1097911063
```

```
----- TxHash ----- TxIx Amount -----  
45a5d30f7d3895591a9645a29385d59436a710d16d3be2b410a25b774bd 0 1274445 Lovelace + 99999 38ed71b759ef679593a85f486554b7ac28f3087802b6bd43f398b2.nedlaparkoin  
9624f8f8e5371429da2f316af547d7cb5402856a80a077f40f39bcc77b3b 0 1487486 Lovelace + 2 6b8607d89639e9413d6637a1a815a7323c69c8abba96e6dbfdb1aa7  
9c9cd19f73c40732148f5078ea48dd8e7926a3f08740ae83d150840f74bf 1 938281283 Lovelace  
b5af310df4ccdfef38cfd9cde40fd7891384ef1e512a58fe577ee29a51069 0 100000000 Lovelace
```

Node: At the time of writing this document, it has both balances: lovelace and asset

Other wise it show will display empty balance:

```
TxHash TxIx Amount
```

```
-----  
----
```

Note: For the main network of Cardano replace “--network-magic 1097911063” with --mainnet in all valid “cardano-cli” commands.

03 Retrieve the current Cardano protocol parameters using this command

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

Display the "protocol.json"

```
# cat protocol.json
{
  "txFeePerByte": 44,
  "minUTxOValue": 1000000,
  "stakePoolDeposit": 500000000,
  "decentralization": 0,
  "poolRetireMaxEpoch": 18,
  "extraPraosEntropy": null,
  "stakePoolTargetNum": 500,
  "maxBlockBodySize": 65536,
  "maxTxSize": 16384,
  "treasuryCut": 0.2,
  "minPoolCost": 340000000,
  "maxBlockExecutionUnits": null,
  "maxBlockHeaderSize": 1100,
  "costModels": {},
  "maxTxExecutionUnits": null,
  "protocolVersion": {
    "minor": 0,
    "major": 4
  },
  "txFeeFixed": 155381,
  "stakeAddressDeposit": 2000000,
  "monetaryExpansion": 3.0e-3,
  "poolPledgeInfluence": 0.3,
  "executionUnitPrices": null,
  "maxValSize": null
}
```

MINTING THE CARDANO ASSET:

Minting the assets on the Cardano network involves the following steps:

01 Create the policy

```
mkdir policy

cardano-cli address key-gen \
  --verification-key-file policy/policy.vkey \
  --signing-key-file policy/policy.skey

touch policy/policy.script && echo "" > policy/policy.script

echo "{" >> policy/policy.script
echo "  \"keyHash\": \"$(./cardano-cli address key-hash
--payment-verification-key-file policy/policy.vkey)\",\" >> policy/policy.script
echo "  \"type\": \"sig\"" >> policy/policy.script
echo "}" >> policy/policy.script
```

Note: The directory name is “policy” and a fresh key pair is required for the policy. Therefore, “policy.vkey” is the public key and the “policy.skey” is the private key.

Display the policy.script

```
../policy# cat policy.script
{
  "keyHash": "e2c3e007410d4a0cb2a6a9c25ecd803ec170a2b5b022cc6fc5f2ebff",
  "type": "sig"
}
```

02 Mint the policy id using the “cardano-cli transaction policyid” command.

```
# ./cardano-cli transaction policyid --script-file ./policy/policy.script
```

Display the policy id:

```
# ./cardano-cli transaction policyid --script-file ./policy/policy.script  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2
```

Note this is the policy id "38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2" and it is generated from the policy.script.

03 Building raw transaction:

"ADA" tokens are required to create transactions from an address. Therefore, 1000 ADA are received from the testnet faucet. Lovelace is the smallest unit of ADA and one ADA is equal to one million Lovelace (1000000).

```
https://explorer.cardano-testnet.iohkdev.io/en/transaction?id=bbafb10df4ccd8fe380cfd0cde40fdf891384af1e512a58fe65f7eea29a55068
```

Cardano is UTXO based, now using this transaction hash id as a transaction input. Name of the asset is "mediaparkcoin". In the transaction one million "mediaparkcoin" is set to mint.

```
# ./cardano-cli transaction build-raw \  
--mary-era \  
--tx-in bbafb10df4ccd8fe380cfd0cde40fdf891384af1e512a58fe65f7eea29a55068#0 \  
--out-file matx.raw \  
--fee 0 \  
--tx-out  
addr_test1vqfnzcx8amctaw2zqmnhrt2ns80n2pqnkjdzcymakj5rnct09mxh+1000000000+"100000  
0  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \  
--mint "1000000  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \  
--minting-script-file ./policy/policy.script
```

Note: ADA has to be set in the lovelace. Whereas the asset amount are allowed without any decimal placing or further units.

Display the raw transaction, it is stored in "matx.raw":

```
# cat matx.raw
{
  "type": "TxBodyMary",
  "description": "",
  "cborHex":
  "84a40081825820bbafb10df4ccd8fe380cfd0cde40fdf891384af1e512a58fe65f7eea29a55068000
  18182
  581d60133160c7eef0beb94206e771ad5381df350413b49a2c349beda541cf821a3b9aca00a1581c38
  ea
  d71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2a14d6d656469617061726b636f696e
  1a0
  00f4240020009a1581c38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2a14d6d6
  564
  69617061726b636f696e1a000f42409f8200581ce2c3e007410d4a0cb2a6a9c25ecd803ec170a2b5b0
  22
  cc6fc5f2ebffffff6f6"
}
```

04 Calculate the minimum fee:

While creating the raw transaction the fee value was set to "0" now using the "cardano-cli transaction calculate-min-fee" command, the transaction fee will be calculated as:

```
./cardano-cli transaction calculate-min-fee \
--tx-body-file matx.raw \
--tx-in-count 1 \
--tx-out-count 1 \
--witness-count 2 \
--testnet-magic 1097911063 \
--protocol-params-file protocol.json

182133 Lovelace
```

The transaction fee is 182133 Lovelace.

05 Build the transaction again:

Build the transaction again using the same parameters except "fee". Total fee of 1812133 Lovelace is set.

Moreover, utxo "bbafb10df4ccd8fe380cfd0cde40fdf891384af1e512a58fe65f7eea29a55068" contains 1000 ADA, which is equal to "1000000000 Lovelace". According to the utxo model of Cardano, the total amount of ADA in the input should be equal to the total amount of ADA. The fee is required for every transaction to be processed so that it can be successful in the blockchain. Therefore, creator of the transaction should follow this:

```
Input amounts = Output amounts
Total received ADA = Fee + Amount changed

Now, compute this, 1000000000 - 182133 = 999817867.
```

So, build the raw transaction according to total fee, sending ADA and receiving back ADA:

```
./cardano-cli transaction build-raw \
--mary-era \
--tx-in bbafb10df4ccd8fe380cfd0cde40fdf891384af1e512a58fe65f7eea29a55068#0 \
--out-file matx.raw \
--fee 182133 \
--tx-out
addr_test1vqfnzcx8amctaw2zqmnhrt2ns80n2pqnkjdzcdymakj5rnct09mxh+999817867+"1000000
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \
--mint "1000000
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \
--minting-script-file ./policy/policy.script
```

06 Sign the transaction:

Sign the transaction using the “cardano-cli transaction sign” command. For this, two secret keys are required: “policy.skey” and “policy/policy.skey”. First secret key belongs to the asset issuer. Second secret key belongs to the policy script which is placed under the directory “policy” and it also contains the same name as “policy.skey”.

```
./cardano-cli transaction sign \  
--signing-key-file policy.skey \  
--signing-key-file policy/policy.skey \  
--testnet-magic 1097911063 \  
--tx-body-file matx.raw \  
--out-file matx.signed
```

07 Submit the transaction:

Submit the transaction to the cardano network using the “cardano-cli transaction submit” command.

```
./cardano-cli transaction submit \  
--tx-file matx.signed \  
--testnet-magic 1097911063  
  
Transaction successfully submitted.
```

The message “Transaction successfully submitted” is returned by the “cardano-cli transaction submit” command. This means that the asset name “mediaparkcoin” has successfully minted with this supply “1000000”.

Verify it by querying the utxo of the address used in the transaction build process.

```
# ./cardano-cli query utxo \  
--address addr_test1vqfnzcx8amctaw2zqgmnhrt2ns80n2pqnkjdzcdymakj5rnct09mxh  
--testnet-magic  
1097911063  
  
TxHash TxIx Amount  
-----  
45a5d36ff7d3895591a9645a629305d594366f710d16d3ba2b4188a25bf7f40d 0 12744445 lovelace + 999999  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin  
9624f8f0e53723429da02f3164f547d7cb54028566a80a077f340f39bccf7b3b 0 1407406 lovelace + 2  
6b8d07d69639e9413dd637a1a815a7323c69c86abbaf66dbfdb1aa7  
9c9cd19f75c40732148f567bea848dd6e7926a3fd8740ae863d15d0410f74bfc 1 958281203 lovelace  
b17a1eba27a2cade5d33b5d8c1b993be287e1b3414842485c4977cb4700e2d6d 0 999817867 lovelace + 1000000  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin
```

The last line with black color is the utxo, which shows that an asset named mediaparkcoin with “1000000” is created. This changed amount is “999817867” and is returned to the same address.

This is the explorer link of this transaction:

```
https://explorer.cardano-testnet.iohkdev.io/en/address?address=addr\_test1vqfnzcx8amctaw2zqgmnhrt2ns80n2pqnkjdzcdymakj5rnct09mxh
```

DESTROYING THE TOKEN:

The minted token e.g. "mediaparkcoin" over the Cardano blockchain can be destroyed. The token destroying process is the same as asset minting. Only the asset "--mint" argument needs to be set to a negative asset amount.

01 Building raw transaction:

```
# ./cardano-cli transaction build-raw \  
--mary-era \  
--tx-in 5bfc0b378456475efc6535bdc339481403e24066aa456bbc4fd5c50ed917e9dc#0 \  
--out-file matx.raw \  
--fee 0 \  
--tx-out  
addr_test1vqfnzcx8amctaw2zqgmhrt2ns80n2pqnkjdzcdymakj5rnct09mxh+12744445+"999999  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \  
--mint "-1 38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin"  
\  
--minting-script-file ./policy/policy.script
```

02 Calculate the minimum fee:

Fee calculation process is the same as used in asset minting.

```
./cardano-cli transaction calculate-min-fee \  
--tx-body-file matx.raw \  
--tx-in-count 1 \  
--tx-out-count 1 \  
--witness-count 2 \  
--testnet-magic 1097911063 \  
--protocol-params-file protocol.json  
  
255555 Lovelace
```

03 Build the transaction again:

Fee is set to 255555.

```
# ./cardano-cli transaction build-raw \  
--mary-era \  
--tx-in 5bfc0b378456475efc6535bdc339481403e24066aa456bbc4fd5c50ed917e9dc#0 \  
--out-file matx.raw \  
--fee 255555 \  
--tx-out  
addr_test1vqfnzcx8amctaw2zqmhrt2ns80n2pqnkjdzcdymakj5rnct09mxh+12744445+"999999  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin" \  
--mint "-1 38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin"  
\  
--minting-script-file ./policy/policy.script
```

04 Sign the transaction:

Signing the transaction for asset destruction is the same process.

```
# ./cardano-cli transaction sign \  
--signing-key-file policy.skey \  
--signing-key-file policy/policy.skey \  
--testnet-magic 1097911063 \  
--tx-body-file matx.raw \  
--out-file matx.signed
```

05 Submit the transaction:

Submitting the transaction is also the same.

```
# ./cardano-cli transaction submit \  
--tx-file matx.signed \  
--testnet-magic 1097911063  
Transaction successfully submitted.
```

This is the explorer link of the asset burned transaction:

```
https://explorer.cardano-testnet.iohkdev.io/en/transaction?id=45a5d36ff7d3895591a9645a629305d594366f710d16d3ba2b4188a25bf7f40d
```

TOKEN TRANSFER:

01 Building raw transaction:

```
./cardano-cli transaction build-raw \  
--mary-era \  
--tx-in 45a5d36ff7d3895591a9645a629305d594366f710d16d3ba2b4188a25bf7f40d#0 \  
--tx-in b17aleba27a2cad5d33b5d8c1b993be287e1b3414842485c4977cb4700e2d6d#0 \  
--out-file ./receivermp1/receiverpm1.raw \  
--fee 0 \  
--tx-out  
addr_test1vrr96xxtzx3m4q7gtavh5uj8t8dlt92phqu0vcentdaktygkk4evj+1012562312+"199999  
9  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin"
```

02 Calculate the minimum fee:

```
./cardano-cli transaction calculate-min-fee \  
--tx-body-file ./receivermp1/receiverpm1.raw \  
--tx-in-count 2 \  
--tx-out-count 1 \  
--witness-count 2 \  
--testnet-magic 1097911063 \  
--protocol-params-file protocol.json  
  
182045 Lovelace
```

03 Build the transaction again:

```
./cardano-cli transaction build-raw \  
--mary-era \  
--tx-in 45a5d36ff7d3895591a9645a629305d594366f710d16d3ba2b4188a25bf7f40d#0 \  
--tx-in b17aleba27a2cad5d33b5d8c1b993be287e1b3414842485c4977cb4700e2d6d#0 \  
--out-file ./receivermp1/receiverpm1.raw \  
--fee 182045 \  
--tx-out  
addr_test1vrr96xxtzx3m4q7gtavh5uj8t8dlt92phqu0vcentdaktygkk4evj+1012380267+"199999  
9  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin"
```

04 Sign the transaction:

```
./cardano-cli transaction sign \  
--signing-key-file policy.skey \  
--testnet-magic 1097911063 \  
--tx-body-file ./receivermp1/receiverpm1.raw \  
--out-file ./receivermp1/receiverpm1.signed
```

05 Submit the transaction:

```
./cardano-cli transaction submit \  
--tx-file ./receivermp1/receiverpm1.signed \  
--testnet-magic 1097911063
```

This is the explorer link of the asset transfer transaction:

```
https://explorer.cardano-testnet.iohkdev.io/en/address?address=addr\_test1vrr96xxtzx3m4q7gtavh5uj8t8dlt92phqu0vcentdaktygkk4evj
```

Also, query this address using the command “cardano-cli query utxo”:

```
TxHash TxIx Amount  
-----  
31a0defc5351e8f396fcd95fa570ff4a18ac5255fdc83fffab8312432b64cfa1 0 1012380267 lovelace + 1999999  
38ead71bf59ef679593685f406554b7ac26f3007802b6bd433f396b2.mediaparkcoin
```