RECORDS KEEPER

D-313, Second Floor,
Village Mohammadpur,
RK Puram, New Delhi – 110066
Cell: +91-9910433954

MultiChain
Private Blockchain Platform

Using Native Assets & Transaction Meta Data

MultiChain allows the creation & tracking of assets t the network level. This is an interesting use case of Blockchain to have defined limited native assets & let peers or nodes transact them among each other in an atomic or non-atomic manner. You can define anything as an asset. A resource-hour, real currency, money in virtual wallet like PayTM in India, CPU hour, Virtual currency like Bitcoin. Let's say for a logistic management company a 1 consignment is one asset. Or one consignment from one city is one asset & they want to create a system to make sure assets/consignments are being transferred from one node to another node (one-office to another office) in such a way that they are not being lost.

So what you want to build is up to you. Possibilities are endless.

Step 1: Make sure you are connected to both the nodes using terminal and are in interactive mode for both the nodes using the below command. Please use your own chain name whatever you have used while creating the chain.

multichain-cli chain1 (To switch to interactive mode for chain1)

```
ubuntu@ip-172-31-50-255:~$ multichain-cli chain1

MultiChain Core RPC client build 1.0 alpha 26 protocol 10006


Interactive mode

chain1:
```

Step 2: Using native assets

RECORDS KEEPER

D-313, Second Floor,
Village Mohammadpur,
RK Puram, New Delhi – 110066
Cell: +91-9910433954

MultiChain supports assets natively at the blockchain level. The identifiers and quantities of assets are encoded within each transaction output, alongside the quantity of the blockchain's native currency (which may be zero if it is not being used). Every MultiChain node tracks and verifies the quantity of assets in transactions, just as it is does with the native currency. Specifically, it checks that the total quantities of all assets in a transaction's outputs are exactly matched by the total in its inputs. MultiChain allows each transaction output can contain any number of different assets.

In MultiChain, assets can be referred to in any of three ways:

- An optional asset name, chosen at the time of issuance. If used, the name must be unique on a blockchain, between both assets and streams. Asset names are stored as UTF-8 encoded strings up to 32 bytes in size and are case insensitive.
- An issuetxid, containing the txid of the transaction in which the asset was issued.
- An assetref which encodes the block number and byte offset of the issuance transaction, along with the first two bytes of its txid.

Ref: MultiChain Website. For more background, please see the MultiChain White Paper.

Now we are going to create a new asset and send it between nodes. On the *first server*, get the address that has the permission to create assets:

```
listpermissions issue
```

So the address with issue permission is 1R9E7GvKHQc6h5jjQ3uxRu1oQxbYBB3bzQ3gWC remember it will be different in your case.

Now we'll create a new asset on this node with 1000 units, each of which can be subdivided into 100 parts, sending it to itself:

```
issue 1R9E7GvKHQc6h5jjQ3uxRu1oQxbYBB3bzQ3gWC asset1 1000 0.01
```

On both servers, verify that the asset named asset1 is listed:

```
listassets
```

Now check the asset balances on each server. The first should show a balance of 1000, and the second should show no assets at all:

```
gettotalbalances
```

On the first server, now try sending 100 units of the asset to the second server's wallet:

```
sendasset 1bDkf8QFnAY3Sf5Fr3iNDYTWvQXNqRAK8gqNts asset1 100
```

D-313, Second Floor,
Village Mohammadpur,
RK Puram, New Delhi – 110066
Cell: +91-9910433954

If and only if you see an error that the address does not have receive permissions. So it's time to add receive and send permissions:

```
grant 1bDkf8QFnAY3Sf5Fr3iNDYTWvQXNqRAK8gqNts receive,send
```

Now try sending the asset again, and it should go through:

```
sendasset 1bDkf8QFnAY3Sf5Fr3iNDYTWvQXNqRAK8gqNts asset1 100
```

If the transaction that created the asset has not yet been confirmed in a block you will see another error and should wait around 30 seconds before trying again. (This assumes you left the block time on the default of 15 seconds.)

Now check the asset balances on each server, including transactions with zero confirmations. They should be 900 and 100 respectively:

```
gettotalbalances 0
```

You can also view the transaction on each node and see how it affected their balances:

```
listwallettransactions 1
```

Step 2: Transaction Meta data

In this section we'll create a transaction that sends 125 units of asset1 along with some metadata. On the Node-1, run:

```
sendwithdata 1bDkf8QFnAY3Sf5Fr3iNDYTWvQXNqRAK8gqNts '{"asset1":125}'
48692066726f6d204d756c7469436861696e21
```

Here 48692066726f6d204d756c7469436861696e21 is the metadata you are sending along with the transaction.

Copy and paste the displayed transaction ID:

Now this transaction can be examined on the second server as below:

```
getwallettransaction <Transaction-id you just got from previous command>
```

RECORDS KEEPER

D-313, Second Floor,
Village Mohammadpur,
RK Puram, New Delhi – 110066
Cell: +91-9910433954

```
chain1: getwallettransaction 985fa82b9ec0e48a917c431e1527b718cf6452c60fb30036f15f31ab41b6290b
{"method":"getwallettransaction","params":["985fa82b9ec0e48a917c431e1527b718cf6452c60fb30036f15f31ab41b6290b"],"id":1,"chain_name":"chain1"}

{
    "balance" : {
        "amount" : 0.00000000,
        "assets" : [
            {
                "name" : "asset1",
                "assetref" : "394-267-7627",
                "qty" : 125.00000000
            }
        ]
    },
    "myaddresses" : [
        "1bDkf8QFnAY3Sf5Fr3iNDYTWvQXNqRAK8gqNts"
    ],
    "addresses" : [
        "1R9E7GvKHQc6h5jjQ3uxRu1oQxbYBB3bzQ3gWC"
    ],
    "permissions" : [
    ],
    "items" : [
    ],
    "data" : [
        "48692066726f6d204d756c7469436861696e21"
    ],
    "confirmations" : 3,
    "blockhash" : "0000fc22c7c46c9801fcac2b9e4aa0289281ff8e7a32078b5e4778956d5e1cdf",
    "blockindex" : 1,
    "blocktime" : 1481267844,
    "txid" : "985fa82b9ec0e48a917c431e1527b718cf6452c60fb30036f15f31ab41b6290b",
    "valid" : true,
    "time" : 1481267841,
    "timereceived" : 1481267841
}
chain1:
```

In the output from this command, you should see the balance field showing the incoming 125 units of asset1 and the data field containing the hexadecimal metadata that was added.

That's it for this lecture.

If you are interested in conducting Blockchain training in your city, office or country please reach out to us using this link or drop us an email at training@recordskeeper.co.