transfer.hpp - Manual Token Transfers Testing

Tests in this scope relate to token transfers into the nft.scribe contract, to ensure they are deposited to the correct account. 1.nft started with 6 TLOS in it from previous transfers.

PASSED – Send 105.0000 TLOS to newly created account with no balance, should pass

⊞ nft.users ▾	C

Query

SCOPE	LOWER_BOUND
eth.mainnet	50

Data Table

#	USERID name	U struct_nft_user
1	1.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"1.nft","wallet":[{"id":1,"token":"6.0000 TLOS"}]}
2	2.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"2.nft","wallet":[]}

Send to 2.nft virtual account

Data Table

#	USERID name	U struct_nft_user
1	1.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"1.nft","wallet":[{"id":1,"token":"6.0000 TLOS"}]}
2	2.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"2.nft","wallet":[{"id":1,"token":"105.0000 TLOS"}]}

PASSED – Send 1.0000 TLOS to zz2.nft, should fail

f(x) transfer ▾ ▶	♡ ⏱

▾ Parameters 👁 Command

FROM

👥 usr1.scribe

TO

👥 nft.scribe

QUANTITY

🗄 1.0000 TLOS

MEMO

zz2.nft@eth.mainnet

▷ **Authorization** usr1.scribe@active

▷ **Ricardian**

▾ **Result** Error

assertion failure with message: User not found.

"code" 500
"message" "Internal Service Error"

This fails because there is a specific username check, even though mathematically zz2 == 2 in the base 31 username system.

PASSED – Send 1,000.0000 TLOS to 1.nft, should pass

#	USERID name	U struct_nft_user
1	1.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"1.nft","wallet":[{"id":1,"token":"1006.0000 TLOS"}]}
2	2.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"2.nft","wallet":[{"id":1,"token":"105.0000 TLOS"}]}

PASSED – Audit of Balances so far

1.nft = 1006 TLOS
2.nft = 105 TLOS
Total: 1111 TLOS

nftservices TABLE – passed

| SERVICE_BALANCE struct_token	TOKENS struct_token[]
{"contract":"name.null...a","id":0,"token":"0.0000 NULL"}	[{"contract":"eosio.token","id":1,"token":"1111.0000 TLOS"}]

tokens TABLE – passed

#	ID uint64	T struct_token
1	1	{"contract":"eosio.token","id":1,"token":"1111.0000 TLOS"}
2	2	{"contract":"tkn2.scribe","id":2,"token":"0.0000 TWO"}
3	3	{"contract":"tkn3.scribe","id":3,"token":"0.0000 THR"}

PASSED – Transfer Fake TLOS tkn1.scribe in of 500 to x.nft, should fail
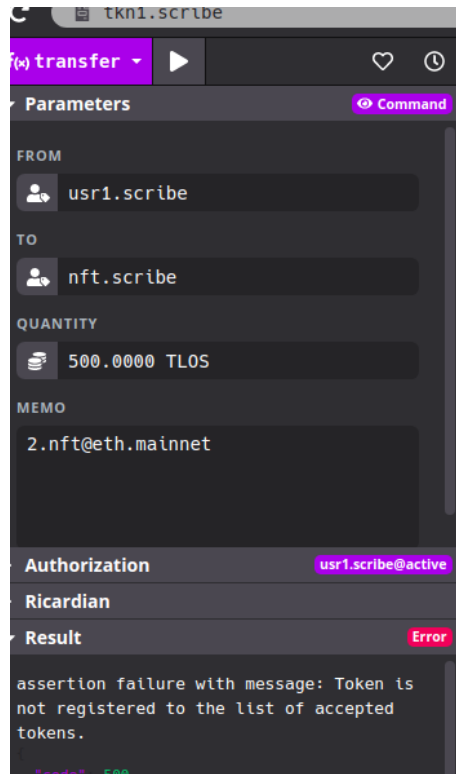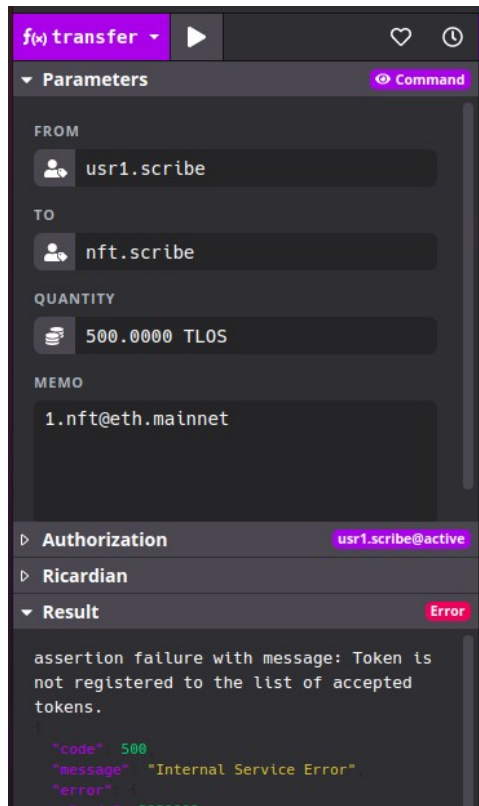
PASSED – Send 500 TLOS to wrong network bsc.mainnet, should fail
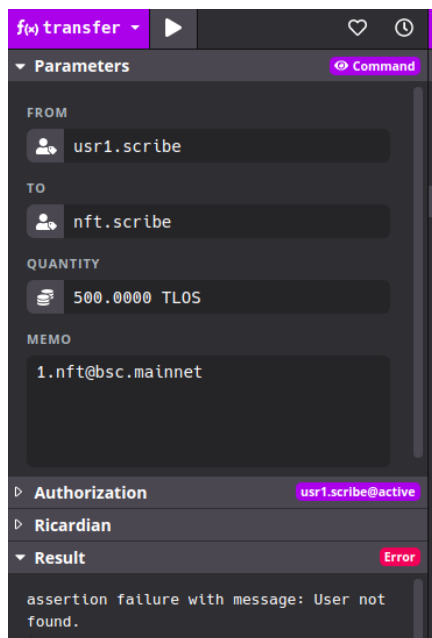
PASSED – Register a new token tkn2.scribe

TOKENS
struct_token[]

[{"contract":"eosio.token","id":1,"token":"1111.0000 TLOS"}]

Parameters   👁 Command

AUTH

👥 usr1.scribe

SUFFIX

👥 nft

NETWORK_ID

👥 eth.mainnet

TOKEN_ID

int 2

▷ **Authorization**   usr1.scribe@active

▷ **Ricardian**

▾ **Result**   Success

Execution succeed. Transaction id:

7ce60628330a110a6439d9dd471fad8a6789223ed5
aafb4538e1fee8819f8ac8

TOKENS
struct_token[]

[{"contract":"eosio.token","id":1,"token":"1111.0000 TLOS"},
{"contract":"tkn2.scribe","id":2,"token":"0.0000 TWO"}]

PASSED – Transfer in 200.0000 TWO to 2.nft, should pass

f(x) transfer ▶

Parameters 👁 Command

FROM
👥 usr1.scribe

TO
👥 nft.scribe

QUANTITY
💰 200.0000 TWO

MEMO
2.nft@eth.mainnet

▷ **Authorization** usr1.scribe@active
▷ **Ricardian**
▼ **Result** Success

Execution succeed. Transaction id:
bcef2652e5b37b458501dbd3fc5b370d18511de66b
5e84e059748e6f2c4d44ba

	SCOPE	LOWER_BOUND	LIMIT
	eth.mainnet	Default: 0	Default: 10

Data Table

#	USERID name	U struct_nft_user
1	1.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"1.nft","wallet":[{"id":1,"token":"1006.0000 TLOS"}]}
2	2.nft	{"a_tps_recent_trx":[],"suffix":"nft","userid":"2.nft","wallet":[{"id":1,"token":"105.0000 TLOS"},{"id":2,"token":"200.0000 TWO"}]}

⊞ tokens ⟳

Query

SCOPE	LOWER_BOUND
nft.scribe	Default: 0

Data Table

#	ID uint64	T struct_token
1	1	{"contract":"eosio.token","id":1,"token":"1111.0000 TLOS"}
2	2	{"contract":"tkn2.scribe","id":2,"token":"200.0000 TWO"}
3	3	{"contract":"tkn3.scribe","id":3,"token":"0.0000 THR"}

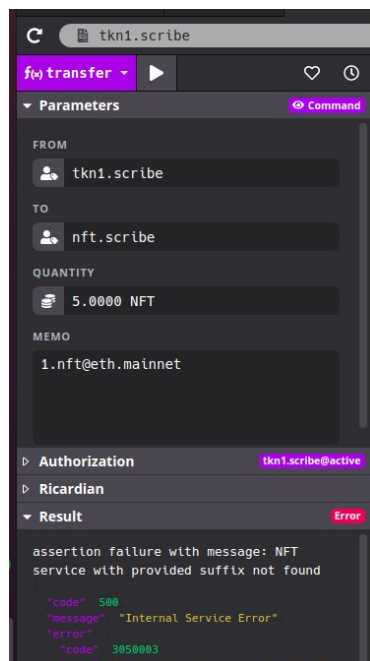**TOKENS**
struct_token[]

S_
str

[{"contract":"eosio.token","id":1,"token":"1111.0000
0 TLOS"},
{"contract":"tkn2.scribe","id":2,"token":"200.0000
TWO"}]

[]

PASSED – Transfer Unregistered Network Token, should fail
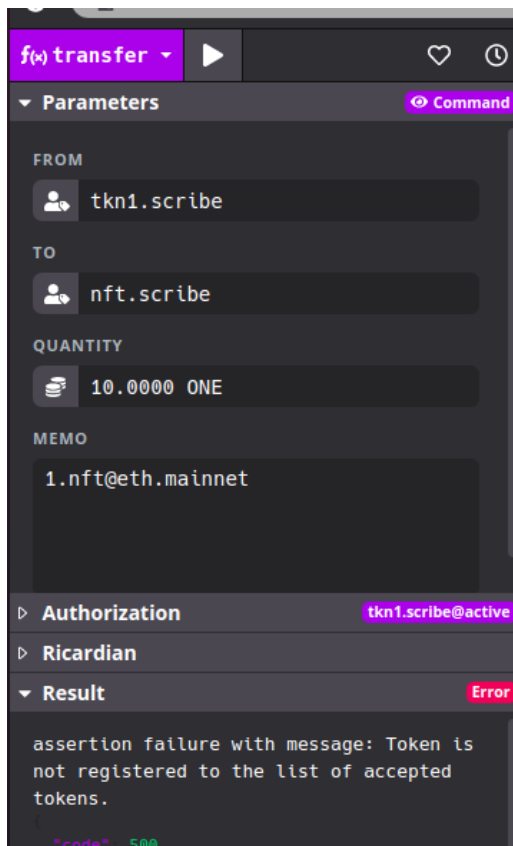
PASSED – Transfer Unregistered NFT Service Token, but registered by admins, should fail

Checks for this are done whenever receiving a token, but not when a token is sent.