The Usage of Qtum Test in SpaceChain OS Environment

The usage of Qtum command line tool

Category	Content
Key words	SpaceChain OS Qtum
Abstract	The usage of Qtum command line tool

Contents

1. Qtumd commands

1.1 Setting Options

Command	Description
-?	Print help information and exit
-version	Print version information and exit
-alertnotify=<cmd>	Execute command when a relevant alert is received or we see a really long fork (%s in cmd is replaced by message)
-blocknotify=<cmd>	Execute the command when the best currency block changes (%s in the command will be replaced with the currency block hash)
-assumevalid=<hex>	If this block is in the chain assume that it and its ancestors are valid and potentially skip their script verification (0 to verify all, default: 000000000000000000013176bf8d7dfeab4e1 db31dc93bc311b436e82ab226b90, testnet: 000000000000128796ee387cf110ccb9 d2f36cffaf7f73079c995377c65ac0dcc)
-conf=<file>	Specify configuration file (default: qtum.conf)
-daemon	Run as a daemon
-datadir=<dir>	Specify data directory
-dbcache=<n>	Set database cache size (in megabytes) (valid values 4 to 16384, default: 450)
-loadblock=<file>	In the startup phase, import blocks from the external blk000**.dat file
-maxorphantx=<n>	Maximum number of unconnected transactions in memory (default: 100)
-maxmempool=<n>	Maximum storage capacity of transaction pool (in MB) (default: 300)
-mempoolexpiry=<n>	Maximum transaction duration in transaction pool (in hours) (default: 336)
-blockreconstructionextratxn=<n>	Extra transactions to keep in memory for compact block reconstructions (default: 100)
-par=<n>	Set the number of script verification threads (-2 to 16, 0 = auto, <0 = leave that many cores free, default: 0)
-pid=<file>	Specify pid file (default: qtumd.pid)
-prune=<n>	Reduce storage requirements by enabling pruning (deleting) of old blocks. This allows the pruneblockchain RPC to be called to delete specific blocks, and enables automatic pruning of

	old blocks if a target size in MiB is provided. This mode is incompatible with -txindex and -rescan. Warning: Reverting this setting requires re-downloading the entire blockchain. (default: 0 = disable pruning blocks, 1 = allow manual pruning via RPC, >550 = automatically prune block files to stay under the specified target size in MiB)
-record-log-opcodes	Logs all EVM LOG opcode operations to the file vmExecLogs.json
-reindex-chainstate	Rebuild chain state from the currently indexed blocks
-reindex	Rebuild chain state and block index from blk*.dat on disk
-sysperms	Create new files with system default permissions, instead of umask 077 (only effective with disabled wallet functionality)
-txindex	Maintain a full transaction index, used by the getrawtransaction rpc call (default: 0)
-logevents	Maintain a full EVM log index, used by searchlogs and gettransactionreceipt rpc calls (default: 0)

1.2 Connection options

Command	Description
-addnode=<ip>	Add a node for connection and try to maintain a connection with that node
-banscore=<n>	The critical value to disconnect from the abnormal node (default: 100)
-bantime=<n>	Number of seconds between disconnecting and reconnecting to an abnormal node (default: 86400)
-bind=<addr>	Bind the specified IP address and port, then listen. Use [host]:port format for IPv6
-connect=<ip>	Only connect to the specified node; -noconnect or -connect=0 disables automatic connection
-discover	Discover own IP addresses (default: 1 when listening and no -externalip or -proxy)
-dns	Allow to query DNS and connect via -addnode, -seednode and -connect (default: : 1)
-dnsseed	Use DNS lookup node (default: 1, ignore -connect/-noconnect)
-externalip=<ip>	Specify your own public address

-forcednsseed	Always query for peer addresses via DNS lookup (default: 0)
-listen	Accept connections from outside (default: 1 if no -proxy or -connect/-noconnect)
-listenonion	Automatically create Tor hidden service (default: 1)
-maxconnections=<n>	Maximum allowable connections (default: 125)
-maxreceivebuffer=<n>	Maximum receive buffer per connection, ×1000 bytes (default: 5000)
-maxsendbuffer=<n>	Maximum send buffer per connection, ×1000 bytes (default: 1000)
-maxtimeadjustment	Maximum allowed median peer time offset adjustment. Local perspective of time may be influenced by peers forward or backward by this amount. (default: 4200 seconds)
-onion=<ip:port>	Use separate SOCKS5 proxy to reach peers via Tor hidden services (default: -proxy)
-onlynet=<net>	Set the type of network that is allowed to connect (ipv4, ipv6 or onion)
-permitbaremultisig	Relay non-P2SH multisig (default: 1)
-peerbloomfilters	Support filtering of blocks and transaction with bloom filters (default: 1)
-port=<port>	Port for listening (default: 8333, testnet: 18333)
-proxy=<ip:port>	Connect via SOCKS5 proxy
-proxyrandomize	Randomize credentials for every proxy connection. This enables Tor stream isolation (default: 1)
-rpcserialversion	Sets the serialization of raw transaction or block hex returned in non-verbose mode, non-segwit(0) or segwit(1) (default: 1)
-seednode=<ip>	Connect to a node to retrieve peer addresses, and disconnect
-timeout=<n>	Set connection timeout (in milliseconds) (Minimum: 1, default: 5000)
-torcontrol=<ip>:<port>	Tor control port to use if onion listening enabled (default: 127.0.0.1:9051)
-torpassword=<pass>	Tor control port password (default: empty)
-dgpstorage	Receiving data from DGP via storage (default: -dgpevm)
-dgpevm	Receiving data from DGP via a contract call (default: -dgpevm)
-whitebind=<addr>	Bind to given address and whitelist peers connecting to it. Use [host]:port notation for IPv6

-whitelist=<IP address or network>	Whitelist peers connecting from the given IP address (e.g. 1.2.3.4) or CIDR notated network (e.g. 1.2.3.0/24). Can be specified multiple times. Whitelisted peers cannot be DoS banned and their transactions are always relayed, even if they are already in the mempool, useful e.g. for a gateway
-whitelistrelay	Accept relayed transactions received from whitelisted peers even when not relaying transactions (default: 1)
-whitelistforcerelay	Force relay of transactions from whitelisted peers even if they violate local relay policy (default: 1)
-maxuploadtarget=<n>	Tries to keep outbound traffic under the given target (in MiB per 24h), 0 = no limit (default: 0)

1.3 Wallet options

Command	Description
-disablewallet	Do not load wallet and disable the RPC call for wallet
-keypool=<n>	Set key pool size (default: 100)
-fallbackfee=<amt>	A fee rate (in BTC/kB) that will be used when fee estimation has insufficient data (default: 0.0002)
-mintxfee=<amt>	Fees (in BTC/kB) smaller than this are considered zero fee for transaction creation (default: 0.00001)
-paytxfee=<amt>	Transaction fee per kilobyte of transaction sent (in BTC/kB) (default: 0.00)
-rescan	In the startup phase, rescan for lost wallet transactions in the block chain
-salvagewallet	Attempt to recover private keys from a corrupt wallet on startup
-spendzeroconfchange	Spend unconfirmed change when sending transactions (default: 1)
-txconfirmtarget=<n>	If paytxfee is not set, include enough fee so transactions begin confirmation on average within n blocks (default: 6)
-usehd	Use hierarchical deterministic key generation (HD) after BIP32. Only has effect during wallet creation/first start (default: 1)
-walletrbf	Send transactions with full-RBF opt-in enabled (default: 0)
-upgradewallet	In the startup phase, upgrade the wallet to the

	latest format
-wallet=<file>	Specify wallet file (in the data directory) (default: wallet.dat)
-walletbroadcast	Enable the wallet to broadcast transaction (default: 1)
-walletnotify=<cmd>	Execute command when a wallet transaction changes (%s in cmd is replaced by TxID)
-zapwallettxes=<mode>	Delete all wallet transactions and only recover those parts of the blockchain through -rescan on startup (1 = keep tx meta data e.g. account owner and payment request information, 2 = drop tx meta data)
-staking =<true/false>	Enables or disables staking (enabled by default)
-stakecache=<true/false>	Enables or disables the staking cache; significantly improves staking performance, but can use a lot of memory (enabled by default)
-rpcmaxgasprice	The max value (in satoshis) for gas price allowed through RPC (default: 100)

1.4 Debug and test options

Command	Description
-uacomment=<cmt>	Append comment to the user agent string
-debug=<category>	Output additional debug information
-help-debug	Display all debug options (usage: --help -help-debug)
-logips	Debug information contains IP address (default: 0)
-logtimestamps	Add time stamp before information debug (default: 1)
-minrelaytxfee=<amt>	Fees (in BTC/kB) smaller than this are considered zero fee for relaying, mining and transaction creation (default: 0.00001)
-maxtxfee=<amt>	Maximum total fees (in BTC) to use in a single wallet transaction or raw transaction; setting this too low may abort large transactions (default: 0.10)
-printtoconsole	Send trace/debug information to console instead of debug.log file
-shrinkdebugfile	Shrink debug.log file on client startup (default: 1 when no -debug)

1.5 Chain selection options

Command	Description
-testnet	Use test chain
-regtest	Use regression test mode and a chain that will be processed immediately and independently. This command is for tool testing and application development

1.6 Node delay options

Command	Description
-bytespersigop	Equivalent bytes per sigop in transactions for relay and mining (default: 20)
-datacarrier	Relay and mine data carrier transactions (default: 1)
-datacarriersize	Maximum size of data in data carrier transactions we relay and mine (default: 83)
-mempoolreplacement	Enable transaction replacement in the memory pool (default: 1)

1.7 Block creation options

Command	Description
-blockmaxweight=<n>	Set maximum BIP141 block weight (default: 3000000)
-blockmaxsize=<n>	Set maximum block size (in bytes) (default: 750000)
-blockprioritysize=<n>	Set high priority/low cost transaction size (in bytes) (default: 0)
-blockmintxfee=<amt>	Set lowest fee rate (in BTC/kB) for transactions to be included in block creation. (default: 0.00001)
-staker-min-tx-gas-price=<amt>	Any contract execution with a gas price below this will not be included in a block (defaults to the value specified by the DGP)
-staker-max-tx-gas-limit=<n>	Any contract execution with a gas limit over this amount will not be included in a block (defaults to soft block gas limit)
-staker-soft-block-gas-limit=<n>	After this amount of gas is surpassed in a block, no more contract executions will be added to the block (defaults to consensus-critical maximum block gas limit)

1.8 RPC service options

Command	Description
-server	Receive command line and JSON-RPC command
-rest	Receive public REST request (default: 0)
-rpcbind=<addr>	Bind to the specified address to listen for JSON-RPC connection. Use [host]:port format for IPv6. This operation can be specified more than once (default: bind to all ports)
-rpccookiefile=<loc>	Specify the address of authorized cookie file (default: data dir)
-rpcuser=<user>	Username for JSON-RPC connection
-rpcpassword=<pw>	Password for JSON-RPC connection
-rpcauth=<userpw>	Username and hashed password for JSON-RPC connection. Format used by <userpw>: <USERNAME>:<SALT>$<HASH>
-rpcport=<port>	Port for JSON-RPC listening (default: 8332, testnet: 18332)
-rpcallowip=<ip>	Allow the specified IP source address to connect to JSON-RPC. The valid <ip> parameter must be a single ip (eg 1.2.3.4) or a single ip with subnet mask (eg 1.2.3.4/255.255.255.0) or an ip with CIDR, (eg 1.2.3.4/24)
-rpcthreads=<n>	Number of threads processing RPC call service (default: 4)

2. Qtum-cli local commands

2.1 Setting Options

Command	Description
-?	Print help information
-conf=<file>	Specify configuration file (default: qtum.conf)
-datadir=<dir>	Specify data path

2.2 Chain selection options

Command	Description
-testnet	Use test chain
-regtest	Use regression test mode and a chain that will be processed immediately and independently. For tool testing and application development
-named	Replace positional parameter by name (default: false)
-rpcconnect=<ip>	Send command to the node on the specified IP (default: 127.0.0.1)
-rpcport=<port>	Specify destination port for JSON-RPC (default: 3889, testnet: 13889)
-rpcwait	Wait for RPC service to start
-rpcuser=<user>	Username for JSON-RPC connection
-rpcpassword=<pw>	Password for JSON-RPC connection
-rpcclienttimeout=<n>	HTTP request timeout (default: 900)
-stdin	Read other parameters from standard input and end a line with EOF/Ctrl-D each time (encrypted transmission is recommended for sensitive information)

3. Qtum-cli RPC commands

3.1 Block commands

Command	Description
callcontract "address" "data" (address)	Get information
getaccountinfo "address"	Get account information via address
getbestblockhash	Return the best block hash in the longest chain
getblock "blockhash" (verbose)	
getblockcount	Get block count
getblockhash height	
getblockheader "hash" (verbose)	
getchaintips	
getdifficulty	Get difficulty
getmempoolancestors txid (verbose)	
getmempooldescendants txid (verbose)	
getmempoolentry txid	
getmempoolinfo	
getrawmempool (verbose)	
getstorage "address"	
gettransactionreceipt "hash"	
gettxout "txid" n (include_mempool)	
gettxoutproof ["txid",...] (blockhash)	
gettxoutsetinfo	
listcontracts (start maxDisplay)	
preciousblock "blockhash"	
pruneblockchain	
searchlogs <fromBlock> <toBlock> (address) (topics)	
verifychain (checklevel nblocks)	
verifytxoutproof "proof"	
waitforlogs (fromBlock) (toBlock) (filter) (minconf)	

3.2 Control commands

Command	Description
getinfo	Get information
getmemoryinfo	Get information about memory usage
help ("command")	See help information for a command
stop	Stop qtum server

3.3 Generating block commands

Command	Description
generate nblocks (maxtries)	Generate block
generatetoaddress nblocks address (maxtries)	

3.4 Mining commands

Command	Description
getblocktemplate (TemplateRequest)	
getmininginfo	
getnetworkhashps (nblocks height)	
getstakinginfo	
getsubsidy [nTarget]	
prioritisetransaction <txid> <priority delta> <fee delta>	
submitblock "hexdata" ("jsonparametersobject")	

3.5 Network commands

Command	Description
addnode "node" "add|remove|onetry"	
clearbanned	
disconnectnode "node"	
getaddednodeinfo ("node")	
getconnectioncount	
getnettotals	
getnetworkinfo	
getpeerinfo	

listbanned	
ping	
setban "subnet" "add|remove" (bantime) (absolute)	
setnetworkactive true|false	

3.6 Transaction commands

Command	Description
createrawtransaction [{"txid":"id","vout":n},...] {"address":amount,"data":"hex", ...} (locktime)	Create a transaction information
decoderawtransaction "hexstring"	Decode the transaction information with signature
sendrawtransaction "hexstring" (allowhighfees)	Send the transaction
signrawtransaction "hexstring" ([{"txid":"id","vout":n,"scriptPubKey": "hex","redeemScript":"hex"},...] ["privatekey1",...] sighashtype)	Sign the transaction information.
decodescript "hexstring"	
fromhexaddress "hexaddress"	
fundrawtransaction "hexstring" (options)	
gethexaddress "address"	
getrawtransaction "txid" (verbose)	

3.7 Tool commands

Command	Description
createmultisig nrequired ["key",...]	Create a transaction information
estimatefee nblocks	Decode the signed transaction information
estimatepriority nblocks	Send the transaction
estimatesmartpriority nblocks	Sign the transaction information
estimatesmartfee nblocks	
signmessagewithprivkey "privkey" "message"	

validateaddress "address"	
verifymessage "address" "signature" "message"	

3.8 Wallet commands

Command	Description
encryptwallet "passphrase"	Encrypt wallet, “passphrase” is the password
walletpassphrase "passphrase" timeout	Decode wallet, “passphrase” is the password, and timeout is the number of seconds it takes for the wallet to be automatically locked again (timer)
walletpassphrasechange "oldpassphrase" "newpassphrase"	Change password, “oldpassphrase” is the original password and “newpassphrase” is the new password
backupwallet "destination"	Create a wallet backup file, "destination" is the backup file name
importwallet "filename"	Load the wallet backup file, "filename" is the backup file name
dumpwallet "filename"	Convert the wallet into human-readable file, "filename" is the filename
getnewaddress	The Bitcoin client maintains an address pool. The size of the address pool can be obtained using the getinfo command and keypoolsize parameter. These addresses are automatically generated and can be used as public receiving addresses or change addresses. Get one of the addresses using getnewaddress command
getreceivedbyaddress “address” minconf	The amount of bitcoin that has been received by the client at the corresponding address can be queried, as well as the number of confirmations required to add this amount to the balance. After sending bitcoin from another wallet for a few seconds, this wallet will respond. “Address” is the specified address, and minconf is the number of confirmations set
getaddressesbyaccount "account"	List all the addresses of the wallet. “Account” is the account and can be an empty character.
getbalance "account" minconf include_watchonly	Display the total balance of all transactions that have been confirmed for at least minconf times
listunspent (minconf maxconf ["addresses",...]	View the unspent amount in the wallet, transaction ID and other information

[include_unsafe])	
getnewaddress ("account")	Get new address from address pool
getwalletinfo	Get wallet information
getaccount "address"	Get account information for the specified address
abandontransaction "txid"	
addmultisigaddress nrequired ["key",...] ("account")	
addwitnessaddress "address"	
bumpfee "txid" (options)	
createcontract "bytecode" (gaslimit gasprice "senderaddress" broadcast)	
dumpprivkey "address"	
getaccountaddress "account"	
getrawchangeaddress	
getreceivedbyaccount "account" (minconf)	
getreceivedbyaddress "address" (minconf)	
gettransaction "txid" (include_watchonly) (waitconf)	
getunconfirmedbalance	
importaddress "address" ("label" rescan p2sh)	
importmulti "requests" "options"	
importprivkey "qtum" ("label") (rescan)	
importprunedfunds importpubkey "pubkey" ("label" rescan)	
keypoolrefill (newsize)	
listaccounts (minconf include_watchonly)	
listaddressgroupings	
listlockunspent	
listreceivedbyaccount (minconf include_empty include_watchonly)	
listreceivedbyaddress (minconf include_empty include_watchonly)	
listsinceblock ("blockhash"	

target_confirmations include_watchonly)	
listtransactions ("account" count skip include_watchonly)	
lockunspent unlock ([{"txid":"txid","vout":n},...])	
move "fromaccount" "toaccount" amount (minconf "comment")	
removeprunedfunds "txid"	
reservebalance [<reserve> [amount]]	
sendfrom "fromaccount" "toaddress" amount (minconf "comment" "comment_to")	
sendmany "fromaccount" {"address":amount,...} (minconf "comment" ["address",...])	
sendmanywithdupes "fromaccount" {"address":amount,...} (minconf "comment" ["address",...])	
sendtoaddress "address" amount ("comment" "comment_to" subtractfeefromamount)	
sendtocontract "contractaddress" "data" (amount gaslimit gasprice senderaddress broadcast)	
setaccount "address" "account"	
settxfee amount	
signmessage "address" "message"	
walletlock	

4. Transaction test

4.1 Make a local transaction

1. Create test coin

Create 500 blocks to generate balance in the account (it may take too long to create 500 blocks at one time, so it can also be created multiple times).

```
./qtum-cli generate –regtest 500
```

Or (set first/root/.qtum/qtum.conf)

```
./qtum-cli -rpcuser=test -rpcpassword=test –regtest generate 500
```

2. Check account balance

```
./qtum-cli getbalance
```

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest getbalance
1019999.99990000
```

```
./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest getbalance
```

3. Get available txid

```
./qtum-cli listunspent
```


```
{
  "txid": "4e564cff32e204fec55305c20950400a595333f5d70d53d7c0b22a840e9ba5f5",
  "vout": 0,
  "address": "qPtnNtX36uhpSnqz59VYHZRuYoS3PwqSTq",
  "scriptPubKey": "2103d9b1f8714e6c3556c9d1836dea4faa8ed2a24be60bdf4f06fac03e4404b7ef6bac",
  "amount": 20000.00000000,
  "confirmations": 503,
  "spendable": true,
  "solvable": true
},
{
  "txid": "eb00e4d56c0d99d2fdc5871091f28e98f9a5225fa10553d35266ce6075f17ffb",
  "vout": 0,
  "address": "qPtnNtX36uhpSnqz59VYHZRuYoS3PwqSTq",
  "scriptPubKey": "2103d9b1f8714e6c3556c9d1836dea4faa8ed2a24be60bdf4f06fac03e4404b7ef6bac",
  "amount": 20000.00000000,
  "confirmations": 549,
  "spendable": true,
  "solvable": true
}
```


4. Get new address

```
./qtum-cli getnewaddress
```

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest getnewaddress
qHxXXv5Kz1XqewPNGYTXN5jbYuyCipabqz
```

5. Create a transaction

Create a transaction using createrawtransaction command and return a string of encrypted hexadecimal strings representing the transaction information.

```
./qtum-cli createrawtransaction

"[{\"txid\":\"eb00e4d56c0d99d2fdc5871091f28e98f9a5225fa10553d35266ce6075f17ffb\",\"vout\":0}]"
"{\"qHxXXv5Kz1XqewPNGYTXN5jbYuyCipabqz\":19999.99990}"
```

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest createrawtransaction "[{\"txid\":\"eb00e4d56c
0d99d2fdc5871091f28e98f9a5225fa10553d35266ce6075f17ffb\",\"vout\":0}]" "{\"qHxXXv5Kz1XqewPNGYTXN5jbYuyCipabqz\":19999.9999}"
0200000001fb7ff17560ce6652d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb000000000ffffffff01f0f849a9d10100001976a91404608f66645ea7
a90046641e9cc48c17590f775088ac00000000
```

6. Sign the transaction information

Use signrawtransaction command to sign the encrypted hexadecimal transaction string and get the signed string.

```
./qtum-cli signrawtransaction

0200000001fb7ff17560ce6652d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb0000000000ffffff
ff01f0f849a9d10100001976a91404608f66645ea7a90046641e9cc48c17590f775088ac00000000
```

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest signrawtransaction 0200000001fb7ff17560ce6652
d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb000000000ffffffff01f0f849a9d10100001976a91404608f66645ea7a90046641e9cc48c17590f7750
88ac00000000
{
  "hex": "0200000001fb7ff17560ce6652d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb0000000048473044022077 8a131af1a595985eac9cc772d1
e4234115f904a0c96ad51a36c53c2eb6bc3702200f8f81f7f5785c33cc10083a915dc36a13959d1ceb00bc621a52cb3a7c12229401ffffffff01f0f849a9d1010000
1976a91404608f66645ea7a90046641e9cc48c17590f775088ac00000000",
  "complete": true
}
```

7. Decode the transaction information

Use decoderawtransaction command to decode the encrypted hexadecimal transaction information and get the transaction details.

```
./qtum-cli decoderawtransaction

0200000001fb7ff17560ce6652d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb0000000048473
0440220778a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc3702200f8f81f7f57
85c33cc10083a915dc36a13959d1ceb00bc621a52cb3a7c12229401ffffffff01f0f849a9d10100001976a914
04608f66645ea7a90046641e9cc48c17590f775088ac00000000
```

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest decoderawtransaction 0200000001fb7ff17560ce66
52d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb000000004847304402207 78a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc
3702200f8f81f7f5785c33cc10083a915dc36a13959d1ceb00bc621a52cb3a7c1222940 1ffffffff01f0f849a9d10100001976a91404608f66645ea7a90046641e9c
c48c17590f775088ac00000000
{
  "txid": "f56013c0c70ca9a68df97b12deced7f1b6b6f1b556f66edb7e7582cfe9ec64d8",
  "hash": "f56013c0c70ca9a68df97b12deced7f1b6b6f1b556f66edb7e7582cfe9ec64d8",
  "size": 157,
  "vsize": 157,
  "version": 2,
  "locktime": 0,
  "vin": [
    {
      "txid": "eb00e4d56c0d99d2fdc5871091f28e98f9a5225fa10553d35266ce6075f17ffb",
      "vout": 0,
      "scriptSig": {
        "asm": "30440220778a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc3702200f8f81f7f5785c33cc10083a915dc36a13959d1c
eb00bc621a52cb3a7c122294[ALL]",
        "hex": "4730440220778a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc3702200f8f81f7f5785c33cc10083a915dc36a13959d
1ceb00bc621a52cb3a7c12229401"
      },
      "sequence": 4294967295
    }
  ],
  "vout": [
    {
      "value": 19999.99990000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_DUP OP_HASH160 04608f66645ea7a90046641e9cc48c17590f7750 OP_EQUALVERIFY OP_CHECKSIG",
        "hex": "76a91404608f66645ea7a90046641e9cc48c17590f775088ac",
        "reqSigs": 1,
        "type": "pubkeyhash",
        "addresses": [
          "qHxXXv5Kz1XqewPNGYTXN5jbYuyCipabqz"
        ]
      }
    }
  ]
}
```

8. Send the transaction information

./qtum-cli sendrawtransaction

0200000001fb7ff17560ce6652d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb0000000048473
0440220778a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc3702200f8f81f7f57
85c33cc10083a915dc36a13959d1ceb00bc621a52cb3a7c12229401ffffffff01f0f849a9d10100001976a914
04608f66645ea7a90046641e9cc48c17590f775088ac00000000

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest sendrawtransaction 0200000001fb7ff17560ce6652
d35305a15f22a5f9988ef2911087c5fdd2990d6cd5e400eb000000004847304402207 78a131af1a595985eac9cc772d1e4234115f904a0c96ad51a36c53c2eb6bc37
02200f8f81f7f5785c33cc10083a915dc36a13959d1ceb00bc621a52cb3a7c1222940 1ffffffff01f0f849a9d10100001976a91404608f66645ea7a90046641e9cc4
8c17590f775088ac00000000
f56013c0c70ca9a68df97b12deced7f1b6b6f1b556f66edb7e7582cfe9ec64d8
```

9. Check account balance again

./qtum-cli getbalance

```
[root@sylixos:/apps/qtum-cli]# ./qtum-cli -rpcuser=test -rpcpassword=test1234 -regtest getbalance
1019999.99980000
```

4.2 Conduct a cross-board transaction

1. Start the respective qtumd

./qtumd -regtest &

2. Gets the address of the receiver

```
./qtum-cli getnewaddress
```

3. Conduct transaction

Same as **错误!未找到引用源。**.

4. Package creation blocks

```
./qtum-cli generate 1
```