RPC _Commands

== Addressindex ==

getaddressbalance

getaddressdeltas

getaddressmempool

getaddresstxids

getaddressutxos

== Blockchain ==

getbestblockhash

getblock "blockhash" (verbosity)

getblockchaininfo

getblockcount

getblockhash height

getblockhashes timestamp

getblockheader "hash" (verbose)

getblockheaders "hash" (count verbose)

getchaintips (count branchlen)

getdifficulty

getmempoolancestors txid (verbose)

getmempooldescendants txid (verbose)

getmempoolentry txid

getmempoolinfo

getrawmempool (verbose)

getspecialtxes "blockhash" (type count skip verbosity)

getspentinfo

gettxout "txid" n (include_mempool)

gettxoutproof ["txid",...] (blockhash)

gettxoutsetinfo

preciousblock "blockhash"

pruneblockchain

verifychain (checklevel nblocks)

verifytxoutproof "proof"

== Control ==

debug "category"

getinfo

getmemoryinfo

help ("command") ("subCommand")

stop

== Evo ==

bls "command" ...

protx "command" ...

quorum "command" ...

== Generating ==

generate nblocks (maxtries)

generatetoaddress nblocks address (maxtries)

== Helleniccoin ==

getgovernanceinfo

getpoolinfo

getsuperblockbudget index

gobject "command"...

masternode "command"...

masternode list ("mode" "filter")

mnsync [status|next|reset]

privatesend "command"

spork "command"

voteraw <masternode-tx-hash> <masternode-tx-index> <governance-hash> <vote-signal> [yes|no|abstain] <time> <vote-sig>

== Mining ==

getblocktemplate (TemplateRequest)

getmininginfo

getnetworkhashps (nblocks height)

prioritisetransaction <txid> <fee delta>

submitblock "hexdata" ("jsonparametersobject")

== Network ==

addnode "node" "add|remove|onetry"

clearbanned

disconnectnode "address"

getaddednodeinfo ("node")

getconnectioncount

getnettotals

getnetworkinfo

getpeerinfo

listbanned

ping

setban "subnet" "add|remove" (bantime) (absolute)

setnetworkactive true|false

== Rawtransactions ==

createrawtransaction [{"txid":"id","vout":n},...] {"address":amount,"data":"hex",...} (locktime)

decoderawtransaction "hexstring"

decodescript "hexstring"

fundrawtransaction "hexstring" (options)

getrawtransaction "txid" (verbose)

sendrawtransaction "hexstring" (allowhighfees instantsend bypasslimits)

signrawtransaction "hexstring" (
[{"txid":"id","vout":n,"scriptPubKey":"hex","redeemScript":"hex"},...] ["privatekey1",...]
sighashtype)

== Util ==

createmultisig nrequired ["key",...]

estimatefee nblocks

estimatesmartfee nblocks

signmessagewithprivkey "privkey" "message"

validateaddress "address"

verifymessage "address" "signature" "message"

== Wallet ==

abandontransaction "txid"

addmultisigaddress nrequired ["key",...] ("account")

backupwallet "destination"

dumphdinfo

dumpprivkey "address"

dumpwallet "filename"

encryptwallet "passphrase"

getaccount "address"

getaccountaddress "account"

getaddressesbyaccount "account"

getbalance ("account" minconf addlocked include_watchonly)

getnewaddress ("account")

getrawchangeaddress

getreceivedbyaccount "account" (minconf addlocked)

getreceivedbyaddress "address" (minconf addlocked)

gettransaction "txid" (include_watchonly)

getunconfirmedbalance

getwalletinfo

importaddress "address" ("label" rescan p2sh)

importelectrumwallet "filename" index

importmulti "requests" "options"

importprivkey "helleniccoinprivkey" ("label") (rescan)

importprunedfunds

importpubkey "pubkey" ("label" rescan)

importwallet "filename"

instantsendtoaddress "address" amount ("comment" "comment-to" subtractfeefromamount)

keepass <genkey|init|setpassphrase>

keypoolrefill (newsize)

listaccounts (minconf addlocked include_watchonly)

listaddressbalances (minamount)

listaddressgroupings

listlockunspent

listreceivedbyaccount (minconf addlocked include_empty include_watchonly)

listreceivedbyaddress (minconf addlocked include_empty include_watchonly)

listsinceblock ("blockhash" target_confirmations include_watchonly)

listtransactions ("account" count skip include_watchonly)

listunspent (minconf maxconf ["addresses",...] [include_unsafe])

lockunspent unlock ([{"txid":"txid","vout":n},...])

move "fromaccount" "toaccount" amount (minconf "comment")

removeprunedfunds "txid"

sendfrom "fromaccount" "toaddress" amount (minconf addlocked "comment" "comment_to")

sendmany "fromaccount" {"address":amount,...} (minconf addlocked "comment" ["address",...] subtractfeefromamount use_is use_ps)

sendtoaddress "address" amount ("comment" "comment_to" subtractfeefromamount use_is use_ps)

setaccount "address" "account"

setprivatesendamount amount

setprivatesendrounds rounds

settxfee amount

signmessage "address" "message"

Command _ Options:

-?

Print this help message and exit

-version

Print version and exit

-alerts

Receive and display P2P network alerts (default: 1)

-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 command when the best block changes (%s in cmd is replaced by 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: 00000e0c09c1e8168627358b3170fb343a5443e12d9443130234c45558d50383, testnet: 000005e2d4cfe29bd3bfe07af45bd9639d284ce3b21ca79b629e7616c895e20a)

-conf=<file>

Specify configuration file (default: helleniccoin.conf)

-datadir=<dir>

Specify data directory

-dbcache=<n>

Set database cache size in megabytes (4 to 1024, default: 300)

-loadblock=<file>

Imports blocks from external blk000??.dat file on startup

-maxorphantx=<n>

Keep at most <n> unconnectable transactions in memory (default: 100)

-maxmempool=<n>

Keep the transaction memory pool below <n> megabytes (default: 300)

-mempoolexpiry=<n>

Do not keep transactions in the mempool longer than <n> 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 (0 to 16, 0 = auto, <0 = leave that many cores free, default: 0)

-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, >945 = automatically prune block files to stay under the specified target size in MiB)

-reindex-chainstate

Rebuild chain state from the currently indexed blocks

-reindex

Rebuild chain state and block index from the blk*.dat files on disk

-txindex

Maintain a full transaction index, used by the getrawtransaction rpc call (default: 1)

-addressindex

Maintain a full address index, used to query for the balance, txids and unspent outputs for addresses (default: 0)

-timestampindex

Maintain a timestamp index for block hashes, used to query blocks hashes by a range of timestamps (default: 0)

-spentindex

Maintain a full spent index, used to query the spending txid and input index for an outpoint (default: 0)

Connection options:

-addnode=<ip>

Add a node to connect to and attempt to keep the connection open

-allowprivatenet

Allow RFC1918 addresses to be relayed and connected to (default: 0)

-banscore=<n>

Threshold for disconnecting misbehaving peers (default: 100)

-bantime=<n>

Number of seconds to keep misbehaving peers from reconnecting (default: 86400)

-bind=<addr>

Bind to given address and always listen on it. Use [host]:port notation for IPv6

-connect=<ip>

Connect only to the specified node(s); -noconnect or -connect=0 alone to disable automatic connections

-discover

Discover own IP addresses (default: 1 when listening and no -externalip or -proxy)

-dns

Allow DNS lookups for -addnode, -seednode and -connect (default: 1)

-dnsseed

Query for peer addresses via DNS lookup, if low on addresses (default: 1 unless -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>

Maintain at most <n> connections to peers (temporary service connections excluded) (default: 125)

-maxreceivebuffer=<n>

Maximum per-connection receive buffer, <n>*1000 bytes (default: 5000)

-maxsendbuffer=<n>

Maximum per-connection send buffer, <n>*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>

Only connect to nodes in network <net> (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>

Listen for connections on <port> (default: 31052 or testnet: 41052)

-proxy=<ip:port>

Connect through SOCKS5 proxy

-proxyrandomize

Randomize credentials for every proxy connection. This enables Tor stream isolation (default: 1)

-seednode=<ip>

Connect to a node to retrieve peer addresses, and disconnect

-timeout=<n>

Specify 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)

-upnp

Use UPnP to map the listening port (default: 0)

-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)

Wallet options:

-disablewallet

Do not load the wallet and disable wallet RPC calls

-keypool=<n>

Set key pool size to <n> (default: 1000)

-fallbackfee=<amt>

A fee rate (in HNC/kB) that will be used when fee estimation has insufficient data (default: 0.00001)

-mintxfee=<amt>

Fees (in HNC/kB) smaller than this are considered zero fee for transaction creation (default: 0.00001)

-paytxfee=<amt>

Fee (in HNC/kB) to add to transactions you send (default: 0.00)

-rescan

Rescan the block chain for missing wallet transactions on startup

-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 BIP39/BIP44. Only has effect during wallet creation/first start (default: 0)

-mnemonic=<text>

User defined mnemonic for HD wallet (bip39). Only has effect during wallet creation/first start (default: randomly generated)

-mnemonicpassphrase=<text>

User defined mnemonic passphrase for HD wallet (BIP39). Only has effect during wallet creation/first start (default: empty string)

-hdseed=<hex>

User defined seed for HD wallet (should be in hex). Only has effect during wallet creation/first start (default: randomly generated)

-upgradewallet

Upgrade wallet to latest format on startup

-wallet=<file>

Specify wallet file (within data directory) (default: wallet.dat)

-walletbroadcast

Make the wallet broadcast transactions (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)

-createwalletbackups=<n>

Number of automatic wallet backups (default: 10)

-walletbackupsdir=<dir>

Specify full path to directory for automatic wallet backups (must exist)

-keepass

Use KeePass 2 integration using KeePassHttp plugin (default: 0)

-keepassport=<port>

Connect to KeePassHttp on port <port> (default: 19455)

-keepasskey=<key>

KeePassHttp key for AES encrypted communication with KeePass

-keepassid=<id>

KeePassHttp id for the established association

-keepassname=<name>

Name to construct url for KeePass entry that stores the wallet passphrase

-windowtitle=<name>

Wallet window title

ZeroMQ notification options:

-zmqpubhashblock=<address>

Enable publish hash block in <address>

-zmqpubhashtx=<address>

Enable publish hash transaction in <address>

-zmqpubhashtxlock=<address>

Enable publish hash transaction (locked via InstantSend) in <address>

-zmqpubhashgovernancevote=<address>

Enable publish hash of governance votes in <address>

-zmqpubhashgovernanceobject=<address>

Enable publish hash of governance objects (like proposals) in <address>

-zmqpubhashinstantsenddoublespend=<address>

Enable publish transaction hashes of attempted InstantSend double spend in <address>

-zmqpubrawblock=<address>

Enable publish raw block in <address>

-zmqpubrawtx=<address>

Enable publish raw transaction in <address>

-zmqpubrawtxlock=<address>

Enable publish raw transaction (locked via InstantSend) in <address>

-zmqpubrawinstantsenddoublespend=<address>

Enable publish raw transactions of attempted InstantSend double spend in <address>

Debugging/Testing options:

-uacomment=<cmt>

Append comment to the user agent string

-debug=<category>

Output debugging information (default: 0, supplying <category> is optional). If <category> is not supplied or if <category> = 1, output all debugging information.<category> can be: addrman, alert, bench, cmpctblock, coindb, db, http, leveldb, libevent, lock, mempool, mempoolrej, net, proxy, prune, rand, reindex, rpc, selectcoins, tor, zmq, helleniccoin (or specifically: chainlocks, gobject, instantsend, keepass, llmq, llmq-dkg, llmq-sigs, masternode, mnpayments, mnsync, privatesend, spork), qt.

-help-debug

Show all debugging options (usage: --help -help-debug)

-logips

Include IP addresses in debug output (default: 0)

-logtimestamps

Prepend debug output with timestamp (default: 1)

-minrelaytxfee=<amt>

Fees (in HNC/kB) smaller than this are considered zero fee for relaying, mining and transaction creation (default: 0.00001)

-maxtxfee=<amt>

Maximum total fees (in HNC) 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 info to console instead of debug.log file

-printtodebuglog

Send trace/debug info to debug.log file (default: 1)

-shrinkdebugfile

Shrink debug.log file on client startup (default: 1 when no -debug)

Chain selection options:

-testnet

Use the test chain

-devnet=<name>

Use devnet chain with provided name

-litemode

Disable all HellenicCoin specific functionality (Masternodes, PrivateSend, InstantSend, Governance) (0-1, default: 0)

-sporkaddr=<helleniccoinaddress>

Override spork address. Only useful for regtest and devnet. Using this on mainnet or testnet will ban you.

-minsporkkeys=<n>

Overrides minimum spork signers to change spork value. Only useful for regtest and devnet. Using this on mainnet or testnet will ban you.

Masternode options:

-masternode

Enable the client to act as a masternode (0-1, default: 0)

-masternodeblsprivkey=<hex>

Set the masternode BLS private key

PrivateSend options:

-enableprivatesend

Enable use of automated PrivateSend for funds stored in this wallet (0-1, default: 0)

-privatesendmultisession

Enable multiple PrivateSend mixing sessions per block, experimental (0-1, default: 0)

-privatesendsessions=<n>

Use N separate masternodes in parallel to mix funds (1-10, default: 4)

-privatesendrounds=<n>

Use N separate masternodes for each denominated input to mix funds (2-16, default: 4)

-privatesendamount=<n>

Keep N HNC anonymized (2-100000000, default: 1000)

-privatesenddenoms=<n>

Create up to N inputs of each denominated amount (10-100000, default: 300)

-liquidityprovider=<n>

Provide liquidity to PrivateSend by infrequently mixing coins on a continual basis (0-100, default: 0, 1=very frequent, high fees, 100=very infrequent, low fees)

InstantSend options:

-enableinstantsend

Enable InstantSend, show confirmations for locked transactions (0-1, default: 1)

-instantsendnotify=<cmd>

Execute command when a wallet InstantSend transaction is successfully locked (%s in cmd is replaced by TxID)

Node relay options:

-bytespersigop

Minimum bytes per sigop in transactions we relay and mine (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)

Block creation options:

-blockmaxsize=<n>

Set maximum block size in bytes (default: 2000000)

-blockmintxfee=<amt>

Set lowest fee rate (in HNC/kB) for transactions to be included in block creation. (default: 0.00001)

RPC server options:

-server

Accept command line and JSON-RPC commands

-rest

Accept public REST requests (default: 0)

-rpcbind=<addr>[:port]

Bind to given address to listen for JSON-RPC connections. This option is ignored unless -rpcallowip is also passed. Port is optional and overrides -rpcport. Use [host]:port notation for IPv6. This option can be specified multiple times (default: 127.0.0.1 and ::1 i.e., localhost, or if -rpcallowip has been specified, 0.0.0.0 and :: i.e., all addresses)

-rpccookiefile=<loc>

Location of the auth cookie (default: data dir)

-rpcuser=<user>

Username for JSON-RPC connections

-rpcpassword=<pw>

Password for JSON-RPC connections

-rpcauth=<userpw>

Username and hashed password for JSON-RPC connections. The field <userpw> comes in the format: <USERNAME>:<SALT>$<HASH>. A canonical python script is included in share/rpcuser. The client then connects normally using the rpcuser=<USERNAME>/rpcpassword=<PASSWORD> pair of arguments. This option can be specified multiple times

-rpcport=<port>

Listen for JSON-RPC connections on <port> (default: 31051 or testnet: 41051)

-rpcallowip=<ip>

Allow JSON-RPC connections from specified source. Valid for <ip> are a single IP (e.g. 1.2.3.4), a network/netmask (e.g. 1.2.3.4/255.255.255.0) or a network/CIDR (e.g. 1.2.3.4/24). This option can be specified multiple times

-rpcthreads=<n>

Set the number of threads to service RPC calls (default: 4)

UI Options:

-choosedatadir

Choose data directory on startup (default: 0)

-lang=<lang>

Set language, for example "de_DE" (default: system locale)

-min

Start minimized

-rootcertificates=<file>

Set SSL root certificates for payment request (default: -system-)

-splash

Show splash screen on startup (default: 1)

-resetguisettings

Reset all settings changed in the GUI