

dune-client Cheatsheet (July 11, 2019)

Access the documentation

man [<keyword>...] [-v - verbosity <0|1|2|3>] [-format <plain|colors|html>] Print documentation of commands.

Global options

-d -base-dir <path> client data directory
-c -config-file <path> configuration file
-t -timings show RPC request times
-chain <hash|tag> chain (main or test)
-b -block <hash|tag> block (head or genesis)
-w -wait <none|<int> confirmation blocks
-p -protocol <hash> use commands of protocol
-l -log-requests log all requests to the node
-A -addr <IP addr|host> IP address of the node
-P -port <number> RPC port of the node
-S -tls use TLS to connect to node.
-R -remote-signer <uri> URI of the remote signer
-f -password-filename <filename> password file

Arguments

[-fee <amount>] [-D -dry-run] FEESARGS
[-force-low-fee] [-verbose-signing]
[-minimal-fees <amount>]
[-minimal-nanodun-per-byte <amount>] [-minimal-nanodun-per-gas-unit <amount>] [-fee-cap <amount>] [-burn-cap <amount>]
[-G -gas-limit <amount>] [-S -storage-limit <amount>] [-C -counter <counter>] [-q -no-print-source] MULTISIGARGS

Binary Description:

describe unsigned block header Describe unsigned block header
describe unsigned operation Describe unsigned block header

Known contracts

remember contract <new> <src> [-f -force] Add a contract to the wallet.
forget contract <name> Remove a contract from the wallet.
list known contracts Lists all known contracts in the wallet.
forget all contracts [-f -force] Forget the entire wallet of known contracts.
show known contract <name> Display a contract from the wallet.

Miscellaneous commands:

list understood protocols List the protocol versions that this client understands.
activate protocol <version> with key <password> and parameters <parameters> Activate a protocol
wait for <operation> to be included [-confirmations <nblocks>] [-check-previous <nblocks>] [-branch <block_hash>] Wait until an operation is included in a block
get receipt for <operation> [-check-previous <nblocks>] Get receipt for past operation
import fundraiser secret key <new> Import a Tezos ICO secret key
activate fundraiser account <new> with <code> Activate a Tezos ICO account

Config

config show Show the config file.
config reset Reset the config file
config update Update the config
config init [-o -output <path>] Create a config file based on the current CLI values

Wallet

list signing schemes List supported signing schemes.
gen keys <new> [-f -force] [-s -sig <ed25519|secp256k1|p256>] [-encrypted] Generate a pair of keys.
gen vanity keys <new> matching [<words>...] [-P -prefix] [-f -force] [-encrypted] Generate keys including the given string.
encrypt secret key Encrypt an unencrypted secret key.
import secret key <new> <uri> [-f -force] Add a secret key to the wallet.
import public key <new> <uri> [-f -force] Add a public key to the wallet.
add address <new> <src> [-f -force] Add an address to the wallet.
list known addresses List all addresses and associated keys.
show address <name> [-S -show-secret] Show the keys associated with an implicit account.
forget address <name> [-f -force] Forget one address.
forget all keys [-f -force] Forget the entire wallet of keys.
generate nonce for <name> from <data> Compute deterministic nonce.
generate nonce hash for <name> from <data> Compute deterministic nonce hash.

RPC

rpc list <uri> List RPCs under uri
rpc schema <HTTP method> <url> Get the input and output JSON schemas of an RPC
rpc format <HTTP method> <url> [-b -binary] Get the humanoid readable input and output formats of an RPC
rpc get <url> RPC with GET method
rpc [post|put] <url> [with <input>] RPC with POST|PUT
rpc delete <url> RPC with DELETE.
rpc get /chains/main/mempool/pending_operations

Ledger Nano S ledger://animal/curve/path

list connected ledgers List devices
show ledger <account-or-ledger> [-test-sign] Display version/public-key/address information for a Ledger URI
get ledger authorized path for <account-or-ledger> Query the path of the authorized key
authorize ledger to bake for <account-or-ledger> Authorize a Ledger to bake for a key (deprecated)
setup ledger to bake for <account-or-ledger> [-main-chain-id <ID>] [-main-hwm <HWM>] [-test-hwm <HWM>] Setup a Ledger to bake for a key
deauthorize ledger baking for <account-or-ledger> Deauthorize Ledger from baking
dune ledger becomes wallet <account-or-ledger> Switch Ledger between wallet and baking
dune ledger becomes baking <account-or-ledger> Switch Ledger between wallet and baking
get ledger high water mark for <account-or-ledger> [-no-legacy-instructions] Get high water mark of a Ledger
set ledger high water mark for <account-or-ledger> to <HWM> Set high water mark of a Ledger

Dune Specifics

dune gen print keys [-s -sig <ed25519|secp256k1|p256>] Generate a pair of keys and print them.
dune hash string <data> [-h -hash <block|operation|public_key|chain_id>] Hash a string.
dune hash file <data> [-h -hash <block|operation|public_key|chain_id>] Hash a file.
dune hexa to json [-i -input <path>] [-o -output <path>] The JSON corresponding to some hexa.
dune generate genesis hash [-d -date <date>] [-o -output <path>] Generate a Genesis block hash.
dune print key hashes <KEY-HASH> Print both dn- and tz- notations for keyhashes

dune-client Cheatsheet

Script Library:

list known scripts	Lists all scripts in the library.
remember script <new> <src> [-f -force]	Add a script to the library.
forget script <name>	Remove script from library.
show known script <name>	Display a library script.
run script <src> on storage <storage> and input <input> [-trace-stack] [-amount <amount>] [-source <source>] [-payer <payer>] [-q -no-print-source] [-G -gas <gas>]	Ask the node to run a script.
typecheck script <src> [-v -details] [-emacs] [-q -no-print-source] [-G -gas <gas>]	Ask the node to typecheck a script.
typecheck data <data> against type <type> [-q -no-print-source] [-G -gas <gas>]	Ask the node to typecheck a data expression.
hash data <data> of type <type> [-G -gas <gas>]	Ask the node to pack a data expression.
unpack michelson data <bytes>	Parse hexa as an expression for Michelson UNPACK.
sign bytes <data> for <src>	Sign bytes for Michelson CHECK_SIGNATURE.
check that <bytes> was signed by <key> to produce <signature> [-q -quiet]	Check the signature of hexa as Michelson CHECK_SIGNATURE.

Validation

bake for <baker> [-max-priority <slot>] [-minimal-fees <amount>] [-minimal-nanodun-per-gas-unit <amount>] [-minimal-nanodun-per-byte <amount>] [-await-late-endorsements] [-f -force] [-minimal-timestamp] [-mempool <file>] [-context <path>]	Forge and inject block using the delegate rights
reveal nonce for [<block_hash>...]	Seed-nonce revelation operation.
reveal nonces	All the possible seed-nonce revelation operations.
endorse for <baker>	Endorsement operation.
filter orphan nonces	Clean the nonces without block
list orphan nonces	List orphan nonces.

On a Given Block

get timestamp [-s -seconds]	Timestamp of the block.
list contracts	Lists all non empty contracts of the block.
get balance for <src>	Balance of a contract.
get script storage for <src>	Storage of a contract.
get big map value for <key> of type <type> in <src>	Get the value associated to a key in the big map storage of a contract.
get script code for <src>	Code of a contract.
get manager for <src>	Manager of a contract.
get delegate for <src>	Delegate of a contract.
set delegate for <src> to <mgr> [FEESARGS]	Set the delegate of a contract.
withdraw delegate from <src> [FEESARGS]	Withdraw the delegate from a contract.
originate account <new> for <mgr> transferring <qty> from <src> [FEESARGS] [-delegate <address>] [-delegatable] [-f -force]	Open a new account.
reveal key for <src> [FEESARGS]	Reveal the public key of the contract manager.
register key <mgr> as delegate [FEESARGS]	Register the public key hash as a delegate.
activate account <new> with <activation_key> [-f -force] [-encrypted]	Register and activate an Testnet faucet account.
originate contract <new> for <mgr> transferring <qty> from <src> running <prg> [FEESARGS] [-S -storage-limit <amount>] [-delegate <address>] [-f -force] [-delegatable] [-spendable] [-init <data>] [-q -no-print-source]	Launch a smart contract on the blockchain.
transfer <qty> from <src> to <dst> [FEESARGS] [-S -storage-limit <amount>] [-C -counter <counter>] [-arg <data>] [-q -no-print-source]	Transfer tokens / call a smart contract.
submit proposals for <delegate> [<proposal>...] [-D -dry-run] [-verbose-signing] [-force]	Submit protocol proposals
submit ballot for <delegate> <proposal> <ballot> [-D -dry-run]	Submit a ballot
show voting period	Summarize current period

Multisig

deploy multisig <new_multisig> for <mgr> transferring <qty> from <src> with threshold <threshold> on public keys [<key>...] [FEESARGS] [-G -gas-limit <amount>] [-S -storage-limit <amount>] [-delegate <address>] [-f -force] [-delegatable] [-spendable] [-q -no-print-source]	Originate a new multisig contract.
prepare multisig transaction on <multisig> transferring <qty> to <dst> [-bytes-only]	Display the threshold, public keys, and byte sequence to sign for a multisigned transfer.
prepare multisig transaction on <multisig> setting delegate to <dlgt> [-bytes-only]	Display the threshold, public keys, and byte sequence to sign for a multisigned delegate change.
prepare multisig transaction on <multisig> withdrawing delegate [-bytes-only]	Display the threshold, public keys, and byte sequence to sign for a multisigned delegate withdraw.
prepare multisig transaction on <multisig> setting threshold to <threshold> and public keys to [<key>...] [-bytes-only]	Display the threshold, public keys, and byte sequence to sign for a multisigned change of keys and threshold.
sign multisig transaction on <multisig> transferring <qty> to <dst> using secret key <key>	Sign a transaction for a multisig contract.
sign multisig transaction on <multisig> setting delegate to <dlgt> using secret key <key>	Sign a delegate change for a multisig contract.
sign multisig transaction on <multisig> withdrawing delegate using secret key <key>	Sign a delegate withdraw for a multisig contract.
sign multisig transaction on <multisig> using secret key <key> setting threshold to <threshold> and public keys to [<key>...]	Sign a change of public keys and threshold for a multisig contract.
from multisig contract <multisig> transfer <qty> to <dst> on behalf of <src> with signatures [<signature>...] [MULTISIGARGS]	Transfer tokens using a multisig contract.
set delegate of multisig contract <multisig> to <dlgt> on behalf of <src> with signatures [<signature>...] [MULTISIGARGS]	Change the delegate of a multisig contract.
withdraw delegate of multisig contract <multisig> on behalf of <src> with signatures [<signature>...] [MULTISIGARGS]	Withdraw the delegate of a multisig contract.
run transaction <bytes> on multisig contract <multisig> on behalf of <src> with signatures [<signature>...] [MULTISIGARGS]	Run a transaction described by a sequence of bytes on a multisig contract.