Subcommands

This reference describes the syntax of the Tessera command line interface (CLI) subcommands.

To start a Tessera node using subcommands, run:

tessera [OPTIONS] [SUBCOMMAND] [SUBCOMMAND OPTIONS]

keygen

Use thekeygen subcommand togenerate one or more key pairs to store in files or a supported key vault.

argonconfig

,keygenconfig

- Syntax
- Example

tessera keygen --argonconfig tessera keygen --argonconfig /home/me/node1/argonoptions.json JSON file containing settings to override thedefault Argon2 configuration .

Legacy syntax for this option is-keygenconfig .

configfile

,config-file

- Syntax
- Example

tessera keygen --configfile tessera keygen --configfile /home/me/me_node/tessera.conf Path to the nodeconfiguration file .

Provide this option whenupdating a configuration file with new keys . Ifconfigout andpwdout are not provided, the updated configuration file prints to the terminal.

configout

- Syntax
- Example

tessera keygen --configout tessera keygen --configfile /home/me/me_node/tessera.conf --configout /home/me/me_node/update/tessera.conf Path to save the updated configuration file to. To use this option, you must supply the--configfile option.

Legacy syntax for this option is-output .

debug

tessera keygen --debug Prints full exception stack traces tostdout .

encryptor.ellipticCurve

- Syntax
- Example

tessera keygen --encryptor.ellipticCurve tessera keygen --encryptor.ellipticCurve secp384r1 Elliptic curve to use for key generation. Defaults tosecp256r1 .

encryptor.nonceLength

- Syntax
- Example

tessera keygen --encryptor.nonceLength tessera keygen --encryptor.nonceLength 38 Nonce length to use as the initialization vector (IV) for symmetric encryption. Defaults to 24.

encryptor.sharedKeyLength

- Syntax
- Example

tessera keygen --encryptor.sharedKeyLength tessera keygen --encryptor.sharedKeyLength 48 Key length to use for symmetric encryption when generating keys. Defaults to 32.

encryptor.symmetricCipher

- Syntax
- Example

tessera keygen --encryptor.symmetricCipher tessera keygen --encryptor.symmetricCipher AES/CTR/NoPadding Symmetric cipher to use for encrypting data. Defaults toAES/GCM/NoPadding .

encryptor.type

- Syntax
- Example

tessera keygen --encryptor.type tessera keygen --encryptor.type EC [Encryption type](#) . Possible values areEC ,NACL , andCUSTOM . Defaults toNACL

keyout

,filename

- Syntax
- Example

tessera keygen --keyout [,...] tessera keygen --keyout /Users/me/keys/nodeKey1,/Users/me/keys/nodeKey2 Comma-separated list of key files to generate. The number of arguments determines the number of key pairs to generate. Defaults tonull .

Legacy syntax for this option is-filename [,...] .

pwdout

- Syntax
- Example

tessera keygen --pwdout tessera keygen --pwdout /home/me/me_node/passwordFile Path to save updated password list to. To use this option, you must supply the[--configout](#) and[--configfile](#) options.

vault.hashicorp.approlepath

- Syntax
- Example

tessera keygen --vault.hashicorp.approlepath tessera keygen --vault.hashicorp.approlepath auth/approle/login [AppRole](#) path for HashiCorp Vault authentication. Defaults toapprole .

Legacy syntax for this option is-keygenvaultapprole .

vault.hashicorp.secretenginepath

- Syntax
- Example

tessera keygen --vault.hashicorp.secretenginepath tessera keygen --vault.hashicorp.secretenginepath /engine/secret Path to the v2 HashiCorp Vault secret engine.

Legacy syntax for this option is-keygenvaultsecretengine .

vault.hashicorp.tlskeystore

- Syntax
- Example

tessera keygen --vault.hashicorp.tlskeystore tessera keygen --vault.hashicorp.tlskeystore /Users/me/auth/keystore.jks Path to JKS keystore for TLS communication with HashiCorp Vault.

Legacy syntax for this option is-keygenvaultkeystore .

vault.hashicorp.tlstruststore

- Syntax
- Example

tessera keygen --vault.hashicorp.tlstruststore tessera keygen --vault.hashicorp.tlstruststore /Users/me/auth/truststore.jks Path to JKS truststore for TLS communication with HashiCorp Vault.

Legacy syntax for this option is-keygenvaulttruststore .

vault.type

- Syntax
- Example

tessera keygen --vault.type tessera keygen --vault.type HASHICORP Key vault provider in which to save the generated key.

If not specified, keys are encrypted and stored on the local filesystem. Valid options areAZURE ,AWS , andHASHICORP .

Legacy syntax for this option iskeygenvaulttype .

vault.url

- Syntax
- Example

tessera keygen --vault.url tessera keygen --vault.url https://secretsmanager.us-west-2.amazonaws.com Key vault base URL.

Legacy syntax for this option is-keygenvaulturl .

keyupdate

,-updatepassword

[Update the password or encryption options](#) for an already locked key, or apply a new password to an unlocked key.

Legacy syntax for this subcommand is-updatepassword [COMMAND OPTIONS] .

configfile

,config-file

- Syntax
- Example

tessera keyupdate --configfile tessera keyupdate --configfile /home/me/me_node/tessera.conf Path to the node's[configuration file](#) .

debug

tessera keyupdate --debug Prints full exception stack traces to stdout .

encryptor.ellipticCurve

- Syntax
- Example

tessera keyupdate --encryptor.ellipticCurve tessera keyupdate --encryptor.ellipticCurve secp384r1 Elliptic curve to use for the updated keys. Defaults to secp256r1 .

encryptor.nonceLength

- Syntax
- Example

tessera keyupdate --encryptor.nonceLength tessera keyupdate --encryptor.nonceLength 38 Nonce length to use as the initialization vector (IV) for symmetric encryption. Defaults to 24.

encryptor.sharedKeyLength

- Syntax
- Example

tessera keyupdate --encryptor.sharedKeyLength tessera keyupdate --encryptor.sharedKeyLength 48 Key length to use for symmetric encryption when updating keys. Defaults to 32.

encryptor.symmetricCipher

- Syntax
- Example

tessera keyupdate --encryptor.symmetricCipher tessera keyupdate --encryptor.symmetricCipher AES/CTR/NoPadding Symmetric cipher to use for encrypting data. Defaults to AES/GCM/NoPadding .

encryptor.type

- Syntax
- Example

tessera keyupdate --encryptor.type tessera keyupdate --encryptor.type EC [Encryption type](#) . Possible values are EC ,NACL , and CUSTOM . Defaults to NACL .

keys.keyData.config.data.aopts.algorithm

- Syntax
- Example

tessera keyupdate --keys.keyData.config.data.aopts.algorithm tessera keyupdate --keys.keyData.config.data.aopts.algorithm id [Argon2](#) variant to use. Defaults to i .

Valid options are i ,d , and id .

keys.keyData.config.data.aopts.iterations

- Syntax
- Example

tessera keyupdate --keys.keyData.config.data.aopts.iterations tessera keyupdate --keys.keyData.config.data.aopts.iterations 4 Number of [Argon2](#) iterations to perform. Defaults to 10.

keys.keyData.config.data.aopts.memory

- Syntax
- Example

tessera keyupdate --keys.keyData.config.data.aopts.memory tessera keyupdate --keys.keyData.config.data.aopts.memory 1248480 Sets the [Argon2](#) memory usage. Defaults to 1048576.

keys.keyData.config.data.aopts.parallelism

- Syntax
- Example

tessera keyupdate --keys.keyData.config.data.aopts.parallelism tessera keyupdate --keys.keyData.config.data.aopts.parallelism 6 Sets the number of parallel [Argon2](#) threads. Defaults to 4.

keys.keyData.privateKeyPath

- Syntax
- Example

tessera keyupdate --keys.keyData.privateKeyPath tessera keyupdate --keys.keyData.privateKeyPath /Users/me/mynode/nodekey.key Path to the private key file to update. This option is required.

keys.password

- Syntax
- Example

tessera keyupdate --keys.passwordFile tessera keyupdate --keys.passwordFile changeme Password to unlock the private key specified using [keys.keyData.privateKeyPath](#) .

keys.passwordFile

- Syntax
- Example

tessera keyupdate --keys.passwordFile tessera keyupdate --keys.passwordFile /Users/me/mynode/passwordFile File containing the password to unlock the private key specified using [keys.keyData.privateKeyPath](#) .

version

`tessera version` Prints version information and exits.