CLI Docs

This tool is a command line interface to Wormhole.

Installation

Pull down the repo if you dont already have it and cd to the appropriate directory:

```

Copy git clone https://github.com/wormhole-foundation/wormhole cd wormhole/clients/js

```

Build and install the cli tool:

```

Copy make install

```

This installs two binaries,worm-fetch-governance andworm on yourPATH .

To useworm , set upHOME/.wormhole/.env with your private keys, based on.env.sample in this folder.

Usage

```

Copy worm

Commands: wormaptosAptosutilities wormedit-vaaEditsorgeneratesaVAA wormevmEVMutilities wormgenerategenerateVAAs(devnetandtestnetonly) worminfoContract,chain,rpcandaddressinformatio nutilities wormnearNEARutilities wormparseParseaVAA(canbeineitherhexorbase64 format) wormrecoverRecoveranaddressfromasignature wormsubmitExecuteaVAA wormsuiSuiutilities wormverify-vaaVerifiesaVAAbyqueryingthecorecontrac tonEthereum

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

```

Subcommands

aptos ```

Copy wormaptos

Commands: wormaptosinit-token-bridgeInittokenbridgecontract wormaptosinit-wormholeInitWormholecorecontract wormaptosdeployDeployanAptospackage wormaptosdeploy-resourceDeployanAptospackageusinga resourceaccount wormaptossend-example-messageSendexamplemessage wormaptosderive-resource-accountDeriveresourceaccountaddress wormaptosderive-wrapped-addressDerivewrappedcointype wormaptoshash-contractsHashcontractbytecodesforupgrade wormaptosupgradePerformupgradeafterVAAhasbeen submitted wormaptosmigratePerformmigrationaftercontract upgrade wormaptosfaucetRequestmoneyfromthefaucetfora givenaccount wormaptosstart-validatorStartalocalaptosvalidator

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

edit-vaa[](https://docs.wormhole.com/wormhole/reference/cli-docs#edit-vaa)

Copy Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] -v,--vaavaainhexformat[string] [required] -n,--networkNetwork [required] [choices:"mainnet","testnet","devnet"] --guardian-set-index,--gsiguardiansetindex[number] --signatures,--sigscommaseparatedlistofsignatures[string] --wormscanurl,--wsuurltowormscanentryforthevaathat includessignatures[string] --wormscan,--wsifspecified,willquerythewormscanentry forthe vaa to get the signatures [boolean] --emitter-chain-id,--ecemitterchainidtobeusedinthevaa [number] --emitter-address,--eaemitteraddresstobeusedinthevaa[string] --nonce,--nononcetobeusedinthevaa[number] --sequence,--seqsequencenumbertobeusedinthevaa[string] --consistency-level,--clconsistencyleveltobeusedinthevaa [number] --timestamp,--tstimestamptobeusedinthevaainunix seconds[number] -p,--payloadpayloadinhexformat[string] --guardian-secret,--gsGuardian's secret key [string]

evm[](https://docs.wormhole.com/wormhole/reference/cli-docs#evm)

Copy wormevm

Commands: wormevmaddress-from-secretComputea20byteethaddressfroma32 byteprivatekey wormevmstorage-updateUpdateastorageslotonanEVMfork duringtesting(anvilorhardhat) wormevmchainsReturnallEVMchains wormevminfoQueryinfoabouttheon-chainstateof thecontract wormevmhijackOverridetheguardiansetofthecore bridgecontractduringtesting(anvil orhardhat) wormevmstart-validatorStartalocalEVMvalidator

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] --rpcRPCendpoint[string]

generate[](https://docs.wormhole.com/wormhole/reference/cli-docs#generate)

Copy wormgenerate[command]

Commands: wormgenerateregistrationGenerateregistrationVAA wormgenerateupgradeGeneratecontractupgradeVAA wormgenerateattestationGenerateatokenattestationVAA wormgeneraterecover-chain-idGeneratearecoverchainIDVAA wormgenerateSetsthedefaultdeliveryprovider set-default-delivery-providerfortheWormholeRelayercontract

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] -g,--guardian-secretGuardians' secret keys (CSV) [string] [required]

info[](https://docs.wormhole.com/wormhole/reference/cli-docs#info)

Copy worminfo[command]

Commands: worminfochain-idPrintthewormholechainIDinteger associatedwiththespecifiedchain name worminfocontractPrintcontractaddress worminfoemitter

Printaddressinemitteraddress format worminfoorigin Printtheoriginchainandaddress oftheassetthatcorrespondstothe givenchainandaddress. worminforegistrationsPrintchainregistrations worminforpcPrintRPCaddress worminfowrappedPrintthewrappedaddressonthe targetchainthatcorrespondswith thespecifiedoriginchainand address.

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

near[](https://docs.wormhole.com/wormhole/reference/cli-docs#near)

Copy wormnear[command]

Commands: wormnearcontract-updateSubmitacontractupdateusingourspecific APIs wormneardeploySubmitacontractupdateusingnearAPIs

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] -m,--moduleModuletoquery[choices:"Core","NFTBridge","TokenBridge"] -n,--networkNetwork[required] [choices:"mainnet","testnet","devnet"] --accountNeardeploymentaccount[string] [required] --attachAttachsomenear[string] --targetNearaccounttoupgrade[string] --mnemonicNearprivatekeys[string] --keyNearprivatekey[string] -r,--rpcOverridedefaultrpcendpointurl[string]

parse[](https://docs.wormhole.com/wormhole/reference/cli-docs#parse)

Copy Positionals: vaavaa[string]

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

recover[](https://docs.wormhole.com/wormhole/reference/cli-docs#recover)

Copy Positionals: digestdigest[string] signaturesignature[string]

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

submit[](https://docs.wormhole.com/wormhole/reference/cli-docs#submit)

Copy Positionals: vaavaa[string]

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] -c,--chainchainname [choices:"unset","solana","ethereum","terra","bsc","polygon", "avalanche","oasis","algorand","aurora","fantom","karura","acala", "klaytn","celo","near","moonbeam","neon","terra2","injective", "osmosis","sui","aptos","arbitrum","optimism","gnosis","pythnet", "xpla","btc","base","sei","rootstock","wormchain","cosmoshub","evmos", "kujira","sepolia"] -n,--networkNetwork [required] [choices:"mainnet","testnet","devnet"] -a,--contract-addressContracttosubmitVAAto(overrideconfig) [string] --rpcRPCendpoint[string] --all-chains,--acSubmittheVAAtoallchainsexceptfortheorigin chainspecifiedinthepayload [boolean] [default:false]

sui[](https://docs.wormhole.com/wormhole/reference/cli-docs#sui)

Copy wormsui

Commands: wormsuibuild-coinBuildwrappedcoinanddumpbytecode.

Example: wormsuibuild-coin-d8-vV__0_1_1-n testnet-r "https://fullnode.testnet.sui.io:443" wormsuideployDeployaSuipackage wormsuiinit-example-message-appInitializeexamplecoremessageapp wormsuiinit-token-bridgeInitializetokenbridgecontract wormsuiinit-wormholeInitializewormholecorecontract wormsuipublish-example-messagePublishmessagefromexampleappvia corebridge wormsuisetup-devnetSetupdevnetbydeployingand initializingcoreandtokenbridgesand submittingchainregistrations. wormsuiobjectsGetownedobjectsbyowner wormsuipackage-idGetpackageIDfromStateobjectID wormsuitxGettransactiondetails

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

transfer[](https://docs.wormhole.com/wormhole/reference/cli-docs#transfer)

Copy Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] --src-chainsourcechain [required] [choices:"solana","ethereum","terra","bsc","polygon", "avalanche","oasis","algorand","aurora","fantom","karura","acala", "klaytn","celo","near","moonbeam","neon","terra2","injective", "osmosis","sui","aptos","arbitrum","optimism","gnosis","pythnet", "xpla","btc","base","sei","rootstock","wormchain","cosmoshub","evmos", "kujira","sepolia"] --dst-chaindestinationchain [required] [choices:"solana","ethereum","terra","bsc","polygon", "avalanche","oasis","algorand","aurora","fantom","karura","acala", "klaytn","celo","near","moonbeam","neon","terra2","injective", "osmosis","sui","aptos","arbitrum","optimism","gnosis","pythnet", "xpla","btc","base","sei","rootstock","wormchain","cosmoshub","evmos", "kujira","sepolia"] --dst-addrdestinationaddress[string] [required] --token-addrtokenaddress[string] [default: native token] --amounttokenamount[string] [required] -n,--networkNetwork[required] [choices:"mainnet","testnet","devnet"] --rpcRPCendpoint[string]

verify-vaa[](https://docs.wormhole.com/wormhole/reference/cli-docs#verify-vaa)

Copy Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean] -v,--vaavaainhexformat[string] [required] -n,--networkNetwork[required] [choices:"mainnet","testnet","devnet"]

status[](https://docs.wormhole.com/wormhole/reference/cli-docs#status)

Copy Positionals: networkNetwork[choices:"mainnet","testnet","devnet"] chainSourcechain [choices:"unset","solana","ethereum","terra","bsc","polygon", "avalanche","oasis","algorand","aurora","fantom","karura","acala", "klaytn","celo","near","moonbeam","neon","terra2","injective", "osmosis","sui","aptos","arbitrum","optimism","gnosis","pythnet", "xpla","btc","base","sei","rootstock","wormchain","cosmoshub","evmos", "kujira","sepolia"] txSourcetransactionhash[string]

Options: --helpShowhelp[boolean] --versionShowversionnumber[boolean]

```

Examples

VAA generation

Usegenerate to create VAAs for testing. For example, to create an NFT bridge registration VAA:

```

Copy wormgenerateregistration--moduleNFTBridge\ --chainbsc\ --contract-address0x706abc4E45D419950511e474C7B9Ed348A4a716c\ --guardian-secretcfb12303a19cde580bb4dd771639b0d26bc68353645571a8cff516ab2ee113a0

```

Example creating a token attestation VAA:

```

Copy wormgenerateattestation--emitter-chainethereum\ --emitter-address11111111111111111111111111111115\ --chainethereum\ --token-address0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48\ --decimals6\ --symbolUSDC\ --nameUSDC\ --guardian-secretcfb12303a19cde580bb4dd771639b0d26bc68353645571a8cff516ab2ee113a0

```

VAA parsing

Useparse to parse a VAA into JSON. For example,

```

Copy worm parse (worm-fetch-governance 13940208096455381020)

```

will fetch governance VAA13940208096455381020 and print it as JSON.

```

Copy

...signatures elided

timestamp:1651416474, nonce:1570649151, emitterChain:1, emitterAddress:'0000000000000000000000000000000000000000000000000000000000000004', sequence:13940208096455381020n, consistencyLevel:32, payload:{ module:'Core', type:'GuardianSetUpgrade', chain:0, newGuardianSetIndex:2, newGuardianSetLength:19, newGuardianSet:['58cc3ae5c097b213ce3c81979e1b9f9570746aa5', 'ff6cb952589bde862c25ef4392132fb9d4a42157', '114de8460193bdf3a2fcf81f86a09765f4762fd1', '107a0086b32d7a0977926a205131d8731d39cbeb', '8c82b2fd82faed2711d59af0f2499d16e726f6b2', '11b39756c042441be6d8650b69b54ebe715e2343', '54ce5b4d348fb74b958e8966e2ec3dbd4958a7cd', '66b9590e1c41e0b226937bf9217d1d67fd4e91f5', '74a3bf913953d695260d88bc1aa25a4eee363ef0', '000ac0076727b35fbea2dac28fee5ccb0fea768e', 'af45ced136b9d9e24903464ae889f5c8a723fc14', 'f93124b7c738843cbb89e864c862c38cddcccf95', 'd2cc37a4dc036a8d232b48f62cdd4731412f4890', 'da798f6896a3331f64b48c12d1d57fd9cbe70811', '71aa1be1d36cafe3867910f99c09e347899c19c3', '8192b6e7387ccd768277c17dab1b7a5027c0b3cf', '178e21ad2e77ae06711549cfbb1f9c7a9d8096e8', '5e1487f35515d02a92753504a8d75471b9f49edb', '6fbebc898f403e4773e95feb15e80c9a99c8348d'] }

```

Submitting VAAs

Usesubmit to submit a VAA to a chain. It first parses the VAA and figures out what's the destination chain and module. For example, a contract upgrade contains both the target chain and module, so the only required argument is the network moniker (mainnet ortestnet ):

```

Copy worm submit (cat my-nft-registration.txt) --network mainnet

```

For VAAs that don't have a specific target chain (like registrations or guardian set upgrades), the script will ask you to specify the target chain. For example, to submit a guardian set upgrade on all chains, simply run:

```

Copy worm-fetch-governance13940208096455381020>guardian-upgrade.txt wormsubmit(catguardian-upgrade.txt)--

```
networkmainnet--chainoasis wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainaurora wormsubmit(catguardian-
upgrade.txt)--networkmainnet--chainfantom wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainkarura
wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainacala wormsubmit(catguardian-upgrade.txt)--networkmainnet--
chainklaytn wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainavalanche wormsubmit(catguardian-upgrade.txt)--
networkmainnet--chainpolygon wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainbsc wormsubmit(catguardian-
upgrade.txt)--networkmainnet--chainsolana wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainterra
wormsubmit(catguardian-upgrade.txt)--networkmainnet--chainethereum wormsubmit(catguardian-upgrade.txt)--
networkmainnet--chaincelo
```

The VAA payload type (guardian set upgrade) specifies that this VAA should go to the core bridge, and the tool directs it there.

info

To get info about a contract (only EVM supported at this time)

```
Copy wormevminfo-cbsc-nmainnet-mTokenBridge

{ "address":"0xB6F6D86a8f9879A9c87f643768d9efc38c1Da6E7",
"wormhole":"0x98f3c9e6E3fAce36bAAd05FE09d375Ef1464288B",
"implementation":"0xEe91C335eab126dF5fDB3797EA9d6aD93aeC9722", "isInitialized":true,
"tokenImplementation":"0xb6D7bbdE7c46a8B784F4a19C7FDA0De34b9577DB", "chainId":4, "governanceChainId":1,
"governanceContract":"0x0000000000000000000000000000000000000000000000000000000000000004",
"WETH":"0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c", "registrations":{
"solana":"0xec7372995d5cc8732397fb0ad35c0121e0eaa90d26f828a534cab54391b3a4f5",
"ethereum":"0x0000000000000000000000003ee18b2214aff97000d974cf647e7c347e8fa585",
"terra":"0x0000000000000000000000007cf7b764e38a0a5e967972c1df77d432510564e2",
"polygon":"0x0000000000000000000000005a58505a96d1dbf8df91cb21b54419fc36e93fde",
"avalanche":"0x0000000000000000000000000e082f06ff657d94310cb8ce8b0d9a04541d8052",
"oasis":"0x0000000000000000000000005848c791e09901b40a9ef749f2a6735b418d7564",
"algorand":"0x0000000000000000000000000000000000000000000000000000000000000000",
"aurora":"0x00000000000000000000000051b5123a7b0f9b2ba265f9c4c8de7d78d52f510f",
"fantom":"0x0000000000000000000000007c9fc5741288cdfdd83ceb07f3ea7e22618d79d2",
"karura":"0x000000000000000000000000ae9d7fe007b3327aa64a32824aaac52c42a6e624",
"acala":"0x000000000000000000000000ae9d7fe007b3327aa64a32824aaac52c42a6e624",
"klaytn":"0x0000000000000000000000005b08ac39eaed75c0439fc750d9fe7e1f9dd0193f",
"celo":"0x000000000000000000000000796dff6d74f3e27060b71255fe517bfb23c93eed",
"near":"0x0000000000000000000000000000000000000000000000000000000000000000",
"injective":"0x0000000000000000000000000000000000000000000000000000000000000000",
"osmosis":"0x0000000000000000000000000000000000000000000000000000000000000000",
"sui":"0x0000000000000000000000000000000000000000000000000000000000000000",
"aptos":"0x0000000000000000000000000000000000000000000000000000000000000000",
"moonbeam":"0x0000000000000000000000000000000000000000000000000000000000000000",
"neon":"0x0000000000000000000000000000000000000000000000000000000000000000",
"terra2":"0xa463ad028fb79679cfc8ce1efba35ac0e77b35080a1abe9bebe83461f176b0a3",
"arbitrum":"0x0000000000000000000000000000000000000000000000000000000000000000",
"optimism":"0x0000000000000000000000000000000000000000000000000000000000000000",
"gnosis":"0x0000000000000000000000000000000000000000000000000000000000000000", } }
```

Misc

To get the contract address for a module:

```
Copy worm info contract mainnet bsc NFTBridge
```

```

To get the RPC address for a chain

```

Copy worm info rpc mainnet bsc

```

Last updated4 months ago

On this page * Installation * Usage * Subcommands * Examples * VAA generation * VAA parsing * Submitting VAAs * info * Misc

Was this helpful? Edit on GitHub
```