Hardhat Verification Plugin

Hardhat is a full-featured development environment for contract compilation, deployment and verification. Thehardhat-verify plugin supports contract verification on BlockScout.

Get Started

1) Install Hardhat

If you are starting from scratch, create an npm project by going to an empty folder, runningnpm init , and following the instructions. Recommend npm 7 or higher.

Once your project is ready:

npm instructions

```
Copy npm install --save-dev hardhat
```

yarn instructions

```
Copy yarn add --dev hardhat
```

2) Create a project

Runnpx hardhat in your project folder and follow the instructions to create (more info here).

3) Install plugin

Install thehardhat-verify plugin

npm

```
Copy npm install --save-dev @nomicfoundation/hardhat-verify
```

yarn

```
Copy yarn add --dev @nomicfoundation/hardhat-verify
```

4) Add plugin reference to config file

Add the following statement to yourhardhat.config.js .

```
Copy require("@nomicfoundation/hardhat-verify");
```

If using TypeScript, add this to yourhardhat.config.ts.More info on using typescript with hardhat available here.

```
Copy import "@nomicfoundation/hardhat-verify";
```

Config File and Unsupported Networks

Your basicHardhat config file (hardhat.config.js orhardhat.config.ts) will be setup to support the network you are working on. In this example we use the Optimism Sepolia test network and a.js file.

Here we add an RPC url without an API key, however some value is still required. You can use any arbitrary stringMore info .

If you prefer, you can migrate tohardhat-toolbox to use a plugin bundle.

In order to use Blockscout explorer for the verification, you have to specify the explorer details under acustomChains object. It includes:

- chainID
-
    - Network chain ID
- apiURL
-
    - Block explorer API URL
- browserURL
-
    - Block explorer URL
-

Find an extensive list of ChainIDs athttps://chainlist.org/ . For example, here we added Blockscout api endpoints for the Optimism Sepolia network to the config file. Note the network name incustomChains must match the network name in theapiKey object.

```

Copy import{ HardhatUserConfig }from"hardhat/config"; import"@nomicfoundation/hardhat-toolbox"; import"@nomicfoundation/hardhat-verify";

constPRIVATE_KEY=vars.get("PRIVATE_KEY");

constconfig:HardhatUserConfig={ solidity:"0.8.24", networks:{ 'optimism-sepolia':{ url:'https://sepolia.optimism.io', accounts:[PRIVATE_KEY], }, }, etherscan:{ apiKey:{ // Is not required by blockscout. Can be any non-empty string 'optimism-sepolia':"abc" }, customChains:[ { network:"optimism-sepolia", chainId:11155420, urls:{ apiURL:"https://optimism-sepolia.blockscout.com/api", browserURL:"https://optimism-sepolia.blockscout.com/", } } ] }, sourcify:{ enabled:false } };

exportdefaultconfig;

```

Deploy and Verify

For deployment we will useHardhat Ignition - built-in Hardhat deployment system.

Deploy

```

Copy

    npx hardhat ignition deploy ./ignition/modules/Lock.ts --network optimism-sepolia ✔ Confirm deploy to network optimism-sepolia (11155420)? … yes Compiled 1 Solidity file successfully (evm target: paris). Hardhat Ignition

Deploying [ LockModule ]

Batch #1 Executed LockModule#Lock

[ LockModule ] successfully deployed

Deployed Addresses

LockModule#Lock - 0xFE826b33e425f99ce962ACB91752DB41F302EFEA

```

Verify

The plugin requires you to include constructor arguments with the verify task and ensures that they correspond to expected ABI signature. However, Blockscout ignores those arguments, so you may specify any values that correspond to the ABI.

```

Copy npx hardhat verify --networkDEPLOYED_CONTRACT_ADDRESS "Constructor argument 1"

```

Optimism Sepolia example.

```

Copy

    npx hardhat verify --network optimism-sepolia 0xFE826b33e425f99ce962ACB91752DB41F302EFEA 1234 Successfully
```

submitted source code for contract contracts/Lock.sol:Lock at 0xFE826b33e425f99ce962ACB91752DB41F302EFEA for verification on the block explorer. Waiting for verification result...

Successfully verified contract Lock on the block explorer. https://optimism-sepolia.blockscout.com/address/0xFE826b33e425f99ce962ACB91752DB41F302EFEA#code

```

Automatically verified contracts

Sometimes the contract may be automatically verified via Ethereum Bytecode Database service. In that case you may see the following response:

```

Copy The contract 0xFE826b33e425f99ce962ACB91752DB41F302EFEA has already been verified on Etherscan. https://optimism-sepolia.blockscout.com/address/0xFE826b33e425f99ce962ACB91752DB41F302EFEA#code

```

In that case, you may try to enforce using--force flag*.

It prevents Hardhat to check if the contract is already verified, and force it to send verification request anyway. Notice, that it is helpful only if the contract was automatically verifiedpartially . That way, a new verification sources would be saved. If the contract wasfully verified already, that just returns an error.

```

Copy npx hardhat verify --networkDEPLOYED_CONTRACT_ADDRESS "Constructor argument 1" --force

```

- The flag is available starting from@nomicfoundation/hardhat-verify@2.0.7
-

Confirm Verification on BlockScout

Go to your BlockScout instance and paste the contract address into the search bar.

?

Scroll down to see verified status. A green checkmark ✅ means the contract is verified.

?

If your screen size is limited, you may need to click the 3 dots to view and click through to the contract.

?

Scroll down to see and interact with the contract code.

?

Video example

Community video created by Carlos Rodriguez for the Soneium Minato testnet, but can be applicable to any chain.

?

FAQ

I am using anOpenZeppelin upgrades plugin implementation and receive an error on proxy contract verification. What should I do? Although you receive an error, the contracts should be verified during the previous steps and you can ignore. Check in the explorer to make sure the contracts have been verified.

Resources

Learn more about plugin configs, troubleshooting etc. athttps://hardhat.org/hardhat-runner/plugins/nomicfoundation-hardhat-verify

Last updated1 month ago
```