# Verifying a smart contract on Voyager

Chen Wen Kang

Follow

Nethermind.eth

--

Listen

Share

With the rapid adoption of Starknet, it has become increasingly important to ensure the security and correctness of smart contracts deployed on the network. The user-friendly voyager-verify

command-line tool makes verifying your Starknet contract hassle-free.

This step-by-step guide will show you how to verify a Starknet contract using the voyager-verify

CLI. There are a few prerequisites for the installation of the CLI package:

- Set up Node.js
- We'll be using npm for this tutorial, but you can use a Node package manager of your choice

Before we continue, make sure you have already built and at least declared your smart contract successfully on chain. You can check if your smart contract is declared by going to https://voyager.online/class/.

Once you have the prerequisites installed, you're good to go! Install the CLI package with:

Now we can start the verification process for your contract. Simply enter the following command to begin the process:

You should see something like this:

As shown above, you will receive a prompt to select the network you wish to verify on. After choosing your network, you'll be prompted for the contract or class address.

You can now input the contract or class address of your contract. Next step — select the Cairo version you're using for your contract.

Once completed, you'll be prompted to select your desired license for your contract - choose whichever best suits your project's needs.

With your license information filled in, you now have to inform the CLI of your contract type: is it an account contract or not?

To complete the verification process, choose the source file you wish to use and provide the name of the contract or class. The class name you provide will appear on the class detail page on Voyager.

You should see a loader indicating that verification of the contract is underway, and it will return a success message if your verification is successful!

# Conclusion

In this guide, we have gone through the steps of verifying a Starknet contract on Voyager using voyager-verify

CLI. Interacting with an unverified contract can be risky, as it's challenging to determine the exact behavior of the contract from its ABI alone. When you verify your smart contracts, you increase transparency for users who interact with your contracts, enabling them to review the code of the contracts they are working with. We hope this tutorial has been helpful and invite you to join our Discord if you have any issues or questions!

# About Us

Nethermind is a team of world-class builders and researchers. We empower enterprises and developers worldwide to access and build upon the decentralized web. Our work touches every part of the web3 ecosystem, from our Nethermind node to fundamental cryptography research and infrastructure for the Starknet ecosystem. Discover our suite of tools for Starknet:

[Warp, the Solidity to Cairo compiler

](https://nethermind.page.link/8a9f),

[Horus, the open-source formal verification tool

](https://nethermind.io/horus/)for Starknet smart contracts,

[Juno

](https://github.com/NethermindEth/juno), Starknet client implementation

, and

[Cairo Smart Contracts Security Auditing

](https://nethermind.page.link/25ee)services. If you're interested in solving some of blockchain's most difficult problems, visit our

[job board

](https://nethermind.page.link/careers)!

# Disclaimer

Kindly note, voyager-verify

CLI is provided on an 'as-is' basis without warranties of any kind. No warranties, express or implied, are being given as to the security of the verified contracts. Please refer to the License Terms for further details.