
neocompiler-eco

Release 2020

NeoResearch Community

Sep 13, 2023

CONTENTS

1	Introduction	3
1.1	What about the Eco?	3
1.2	An Online Laboratory	3
1.3	Acknowledgements	4
1.4	Contributors	4
1.5	License	4
2	Installation	5
2.1	Cloning from GitHub	5
2.2	Linux Installation	5
3	Quick start	7
3.1	Hello World on C#	7
4	Concepts	9
4.1	Shared PrivateNet	9
5	License	11
6	Indices and tables	13

Welcome to NeoCompiler-Eco Documentation, please select the desired content section.

INTRODUCTION

NeoCompiler-Eco project started in early 2018, seeking to provide easy online access to [Neo Blockchain](#) compiling suite for C# [Smart Contracts](#), also called *NeoContract*. The project evolved quickly by supporting other popular languages for NeoContract, such as Python, Java and Go, and also supporting online testing network, a concept called [Shared PrivateNet](#).

The first version was published online on <https://neocompiler.io>, which is still under frequent maintenance up to 2020. The product NeoCompiler-Eco is developed and maintained by [NeoResearch Community](#), including many collaborators from the [GitHub](#) project.

1.1 What about the Eco?

The *Eco* is an abbreviation for *ecosystem*, as the platform itself is able to support the whole development cycle of a smart contract, including: compiling, deploying and testing.

Hint: Although not recommended, the online platform is also capable of deploying and executing *real* transactions on a *real network* (such as MainNet or TestNet). A more secure approach is to do this [locally on your own computer](#).

1.2 An Online Laboratory

More than a development tool for smart contracts, NeoCompiler-Eco platform supports many testing tools for Neo Blockchain itself, named *EcoLab*:

- Message monitoring and timing analysis for [dBFT](#) consensus
- Disassembly for [NeoVM](#) operations (*opcodes*)
- Creation and submission of arbitrary transactions
- Monitoring of storage states for Neo Blockchain, including latest NeoX cross-chain operations

Hint: All of this can also be done in **your machine**! Just [clone the repository](#) and run things locally.

1.3 Acknowledgements

Neo Foundation has supported the project since its early phases, and without that it wouldn't be possible! A special thanks goes to all dozens of developers that fixed small bugs and gave amazing suggestions to make this platform better.

NeoCompiler-Eco is *open-source* and made with for Neo Ecosystem.

1.4 Contributors

Several contributors made this possible, please refer to the [NeoCompiler-Eco project on GitHub](#).

This project is maintained by [NeoResearch Community](#).

1.5 License

Project is free, and source code is released under MIT license.

See complete [license](#).



NeoResearch Logo



NeoResearch First Logo (Historical)

INSTALLATION

Please follow instructions from <https://github.com/neoresearch/neocompiler-eco>.

2.1 Cloning from GitHub

To clone OptFrame repository from GitHub:

```
git clone https://github.com/neoresearch/neocompiler-eco.git
```

2.2 Linux Installation

After cloning, you can build everything:

```
./build_everything.sh
```

This is likely to *take a while*, if you haven't downloaded any of the docker containers yet.

Warning: This process will consume several gigabytes in disk for docker images and dependencies.

2.2.1 Why does it take so long?

Local installation will download all available compiling suite (C#, Python, Go, ...) and also setup a local privatenet for testing.

QUICK START

3.1 Hello World on C#

```
// importing NeoContract for C#
using Neo.SmartContract.Framework.Services.Neo;
// Developing a basic "Hello World" smart contract, or "Hello Visitor1234"
namespace Neo.SmartContract
{
    public class HelloWorld : Framework.SmartContract
    {
        public static void Main()
        {
            Storage.Put("Hello", "Visitor1234");
        }
    }
}
```

Since we are dealing with an online platform with a [Shared PrivateNet](#), it is better to personalize your own contract, by changing *Visitor1234* with your preferred name.

To compile it and see the results, click on *Compile*.

CONCEPTS

We present some general concepts regarding blockchain technology, specifically regarding Neo2.

4.1 Shared PrivateNet

This proposal is unique and powerful, in the sense that it expands local testing (called *PrivateNet*) to a global scale (thus allowing interaction with other developers worldwide in real time). It contrasts with the official *MainNet* in the sense that assets are not *real*, which is also the case for the official *TestNet*. One difference from *TestNet* is that no *faucet* is required, since development assets are refreshed from time to time (previously 12 hours, then 7 days), allowing quick and frequent upgrades from Neo core development branches (always with bleeding edge innovations) and high availability.

Danger: This section is incomplete!

LICENSE

Project is released under MIT license.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`