Installing the circom ecosystem

⚠ Important deprecation note

The oldcircom compiler written in Javascript will be frozen, but it can still be downloaded from theold circom repository .

Installing dependencies

You need several dependencies in your system to runcircom and its associated tools.

- The core tool is thecircom
- compiler which is written in Rust.
- To have Rust available in your system, you can installrustup
- . If you're using Linux or macOS, open a terminal and enter the following command:

TODO remove the command and put a link to rustup sitecurl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh * We also distribute a series of npm packages soNode.js * and some package manager likenpm * oryarn * should be available in your system. Recent versions ofNode.js * include big integer support and web assembly compilers that help run code faster, so to get a better performance, install version 10 or higher.

Installing circom

To install from our sources, clone thecircom repository:

git clone https://github.com/iden3/circom.git Enter the circom directory and use the cargo build to compile:

cargo build --release The installation takes around 3 minutes to be completed. When the command successfully finishes, it generates thecircom binary in the directorytarget/release . You can install this binary as follows (Note : Make sure you're still in the circom directory when running this command) :

cargo install --path circom The previous command will install thecircom binary in the directoryHOME/.cargo/bin .

Now, you should be able to see all the options of the executable by using thehelp flag:

circom --help

circom compiler 2.1.7 IDEN3 Compiler for the circom programming language

USAGE: circom [FLAGS] [OPTIONS] [--] [input]

FLAGS: --r1cs Outputs the constraints in r1cs format --sym Outputs witness in sym format --wasm Compiles the circuit to wasm --json Outputs the constraints in json format --wat Compiles the circuit to wat -c, --c Compiles the circuit to c --O0 No simplification is applied --O1 Only applies signal to signal and signal to constant simplification --O2 Full constraint simplification --verbose Shows logs during compilation --inspect Does an additional check over the constraints produced --use_old_simplification_heuristics Applies the old version of the heuristics when performing linear simplification --simplification_substitution Outputs the substitution applied in the simplification phase in json format -h, --help Prints help information -V, --version Prints version information

OPTIONS: -o, --output Path to the directory where the output will be written [default: .] -p, --prime To choose the prime number to use to generate the circuit. Receives the name of the curve (bn128, bls12381, goldilocks, grumpkin, secq256r1, pallas, vesta) [default: bn128] -l ... Adds directory to library search path --O2round Maximum number of rounds of the simplification process

ARGS: Path to a circuit with a main component [default: ./circuit.circom]

Installing snarkjs

snarkjs is a npm package that contains code to generate and validate ZK proofs from the artifacts produced bycircom .

You can installsnarkjs with the following command:

npm install -g snarkjs