

# Post Processing Tools

The size of the contract is a critical characteristic. The best way to keep it small is a well-designed minimalistic code with a reduced number of dependencies. It is especially important for large contracts and huge multi-contract dApps that can take a fortune to deploy.

When you have done your best with the code optimization it is worth reducing the size of the contract by minifying it.

## Ready to use script

We have prepared a simplebash script that can be used to minify.wasm contract file. You can find [here](#) .

The current approach to minification is the following:

1. Snip (i.e. just replace with unreachable instruction) few known fat functions from the standard library (such as float formatting and panic-related) withwasm-snip
2. .
3. Runwasm-gc
4. to eliminate all functions reachable from the snipped functions.
5. Strip unneeded sections, such as names withwasm-strip
6. .
7. Runbinaryen wasm-opt
8. , which cleans up the rest.

## Requirements to run the script:

- install[wasm-snip](#)
- and[wasm-gc](#)
- with Cargo:

cargo install wasm-snip wasm-gc \* instal[binaryen](#) \* and[wabt](#) \* on your system. For Ubuntu and other Debian based Linux distributions run:

apt install binaryen wabt

## WARNING

Minification could be rather aggressive, so you must test the contract after minification. Standalone NEAR runtime could be helpful[here](#) . [Edit this page](#) Last updatedonAug 24, 2022 byDamián Parrino Was this page helpful? Yes No

[Previous Rapid Prototyping](#) [Next Reproducible Builds](#)