Troubleshooting installation from source

Not enough RAM

Building binaries locally is a computationally heavy task and will put your computer to the test. The compilation usually requires at least 16 GB of RAM and depending on the optimisation of your machine, could require slightly more (for some machines slightly less). For this reason, compilation can sometimes fail.

The error

src/apps/namada lib could not compile due to previous errors. Exited with exit code: is a common error that can sometimes mean your computer ran out of memory when compiling. To resolve this, I have found closing all other applications and recompiling once or twice will do the trick. Otherwise more RAM will be needed.

Compiling for the first time

Compilation errors due to missing library installations when building the binaries for the first time can be a common problem.

Linker "CC" not found

If one runs into the error

Entering directory '/root/namada/wasm/wasm_source' RUSTFLAGS='-C link-arg=-s' cargo build --release --target wasm32-unknown-unknown --target-dir 'target' --features tx_bond && \ cp "./target/wasm32-unknown-unknown/release/namada_wasm.wasm" ../tx_bond.wasm Compiling proc-macro2 v1.0.46 Compiling quote v1.0.21 error: linker cc not found | = note: No such file or directory (os error 2)

error: could not compile quote due to previous error warning: build failed, waiting for other jobs to finish... error: could not compile proc-macro2 due to previous error It may be resolved by running

sudo

apt

install

build-essential Another solution can sometimes be installinglibcland-dev . This can be achieved through:

sudo

apt-get

update

-y sudo

apt-get

install

-y

libclang-dev

WASM32-unknown-unknown

Another issue the compiler may run into is that it cannot find thewasm32-unknown-unknown target.

error[E0463]: can't find crate for core | = note: the wasm32-unknown-unknown target may not be installed = help: consider downloading the target with rustup target add wasm32-unknown-unknown

error[E0463]: can't find crate for compiler_builtins

For more information about this error, try rustc --explain E0463. error: could not compile cfg-if due to 2 previous errors This issue can be resolved by running

rustup

target

add

wasm32-unknown-unknown (Yes the name of the target iswasm32-unknown-unknown . This is not the compiler unable to tell which version/release it is).

OpenSSL

If you run into the error

Could not find directory of OpenSSL installation, and this-sys crate cannot proceed without this knowledge. If OpenSSL is installed and this crate had trouble finding it, you can set the OPENSSL_DIR environment variable for the compilation process.

Make sure you also have the development packages of openssl installed. For example,libssl-dev on Ubuntu or openssl-devel on Fedora.

If you're in a situation where you think the directory*should* be found automatically, please open a bug at https://github.com/sfackler/rust-openssl and include information about your system as well as this message. Then the solution is spelled out for you. You need to install the development packages of OpenSSL. For Ubuntu, this islibssl-dev . For Fedora, this isopenssl-devel . For other distributions, please refer to theOpenSSL website(opens in a new tab) .

For Ubuntu, this can be achieved through

sudo

apt-get

install

libssl-dev

Pre-requisites Binaries