

Installing Rust

IEEE42069 Introduction to Fe_2O_3 / $\text{Fe}(\text{OH})_3$

Tan Hong Kai

IEEE UNM

IEEE Workshop

Installing rustup for Windows

① Go To:

<https://www.rust-lang.org/tools/install>

Installing rustup for Windows

- 1 Go To:
<https://www.rust-lang.org/tools/install>
- 2 Download the relevant 'rustup-init.exe' file (32-bit or 64-bit)

Installing rustup for Windows

❶ Go To:

<https://www.rust-lang.org/tools/install>

❷ Download the relevant 'rustup-init.exe' file (32-bit or 64-bit)

❸ Install Visual Studio build tools if needed.

<https://visualstudio.microsoft.com/downloads/>

Work load to include when prompted:

- 'Desktop Development with C++'
- The Windows 10 or 11 SDK
- The English language pack component, along with any other language pack of your choosing

Installing rustup for Windows

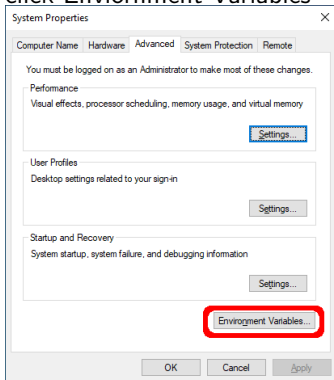
- ④ Adding Rust to PATH Environment Variable (might not be needed)
All rust tools are installed in `C:/Users/username/.cargo/bin`

Installing rustup for Windows

- 4 Adding Rust to PATH Environment Variable (might not be needed)

All rust tools are installed in `C:/Users/username/.cargo/bin`

- 5 Go to System Properties and click Environment Variables

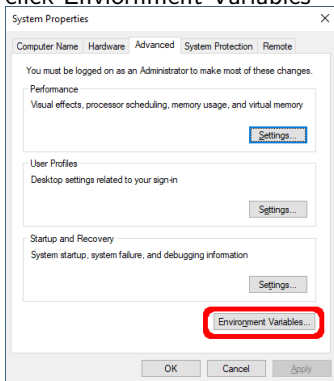


Installing rustup for Windows

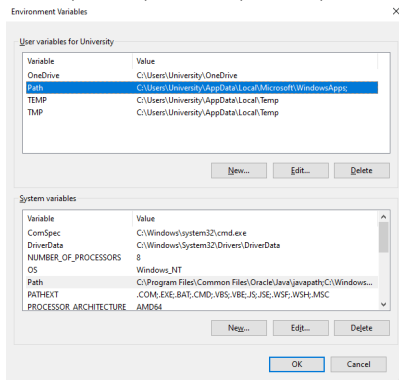
- 4 Adding Rust to PATH Environment Variable (might not be needed)

All rust tools are installed in `C:/Users/username/.cargo/bin`

- 5 Go to System Properties and click Environment Variables



- 6 Add the respective PATH to it
e.g. `C:/Users/SUSKE/.cargo/bin`



Installing rustup for Unix Like (Linux/macOS) Operating Systems

① Open up a terminal

② Run:

```
$ curl -proto 'https' -tlsv1.3 https://sh.rustup.rs -sSf  
| sh
```

③ Follow the instructions

④ Making sure a linker is installed

On MacOS:

```
$ xcode-select -install
```

On Linux, follow distribution documentation instructions on installing GCC or Clang:

On Ubuntu:

```
$ sudo apt update && sudo apt upgrade
```

```
$ sudo apt install build-essential
```


Testing Installation

To check whether Rust is installed correctly run in any CLI e.g. PowerShell and CMD on windows or terminal on Unix:

```
$ rustc --version
```

If the installation is successful, the output of the command should appear something like this:

```
rustc x.y.z (abcabcabc yyyy-mm-dd)
```

If it didn't work double check if the `PATH` variable is setup correctly.

Your First Rust Code

Here is a simple example of a Rust code that prints “Hello World”

```
1 fn main() {  
2     println!("Hello World");  
3 }
```

The file extension of Rust source code file is `.rs`. Save this file as `hello_world.rs`.

The follow sections will guide you through the work flow of compiling and running Rust programs.

Compiling Using rustc

The `rustc` binary is the compiler of Rust. To compile the a Rust code simply run:

```
1 $ rustc hello_world.rs
```

A new binary file should be created with the same name as your source code file. In this case it is `hello_world`. Now test out the binary.

```
1 $ ./hello_world  
2 Hello World
```

Note: If you are compiling on other operating systems the resultant binary might have extensions such as `.exe` on Windows. However, the resultant binary should still have the same behavior.

Compiling Using Cargo

Normally Rust programmers will start new projects using cargo. Cargo is Rust's package manager like apt on Ubuntu/Debian. You can also use it to create your own package and build Rust code.

```
1 $ cargo new hello_world
2 $ cd hello_world
3 $ cargo build
```

The binary should be created in the path `target/debug/packagename`. You can just run the built binary file or you can run it using cargo.

```
1 $ cargo run
```

`cargo run` will build the binary if any changes is made to the source code and run the resultant binary.

IDE

An Integrated Development Environment (IDE). In layman's term a fancy text editor that helps you code.

`rust-analyzer` is a tool made by the Rust community to help with IDE integration. It speaks Language Server Protocol (LSP) that helps IDE and programming languages to communicate with each other.

Website Homepage: <https://rust-analyzer.github.io/>

The rust-analyzer binary needs to be installed for most editor to interact with it (not needed for VS Code). The binary is available in `rustup`.

```
1 $ rustup component add rust-analyzer
```

Note: It might not be installed in `.cargo/bin` directory. Please refer to `rust-analyzer` [manual](#) for more information.

Rust mode & Rustic mode

`rust-mode` is a mode created to help write Rust code in Emacs.

`rustic-mode` is a superset of `rust-mode`, it provides additional features to `rust-mode` such as cargo popup and automatic LSP configuration.

To install paste this into your `init.el`:

```
1 (require 'package)
2 (setq package-archives '(("melpa" . "http://melpa.org/
   packages/")
3                           ("gnu" . "http://elpa.gnu.org/
   packages/")))
4 (package-initialize)
5 (package-refresh-contents)
6 (use-package rustic)
```

Note: The `use-pacakge` and `rustic` package needs to be **installed**.

Rustic Keybindings/Commands

C-c C-c C-u : Compile your current project using cargo.

C-c C-c C-r : Runs `cargo run`.

C-c C-c a : Adds a new crate to the project's Cargo.toml

C-c C-c r : Removes a crate from the project's Cargo.toml

C-c C-c C-f : Formats your current buffer using `rustfmt`.

C-c C-c d : Generates documentation for the current cargo project.

C-c C-c C-t : Runs `cargo test` for the current project. (Running Tests)

C-c C-p : Shows a popup window that runs commands.

More commands are documented in rustic [GitHub README.md](#).

Visual Studio Code Plugin (Cringe)

The IDE integration for VS Code is done by installing the `rust-analyzer` plugin in the [Visual Studio Code marketplace](#).

- 1 Go to the extensions tab (Ctrl + Shift + X)

For more information on how to install extensions in VS Code, visit the VS Code [documentation](#).

Visual Studio Code Plugin (Cringe)

The IDE integration for VS Code is done by installing the `rust-analyzer` plugin in the [Visual Studio Code marketplace](#).

- 1 Go to the extensions tab (Ctrl + Shift + X)
- 2 Search for rust-analyzer

For more information on how to install extensions in VS Code, visit the VS Code [documentation](#).

Visual Studio Code Plugin (Cringe)

The IDE integration for VS Code is done by installing the `rust-analyzer` plugin in the [Visual Studio Code marketplace](#).

- 1 Go to the extensions tab (Ctrl + Shift + X)
- 2 Search for rust-analyzer
- 3 Hit install on the rust-analyzer extension

For more information on how to install extensions in VS Code, visit the VS Code [documentation](#).

Additional Resources I

The Rust Programming Language Book

Rustlings

Rust By Example

Rust Tools

Rust Playground

Rust Documentation

- Local Installation

```
$ rustup doc
```

- Online Documentation

IDE With Top Class Support

- Emacs
- VS Code
- Vim
- Sublime Text
- Atom
- IntelliJ Idea
- Eclipse
- Geany