

Appendix F - Useful Development Tools

In this appendix, we talk about some useful development tools that the Cairo project provides. We'll look at automatic formatting, quick ways to apply warning fixes, a linter, and integrating with IDEs.

Automatic Formatting with `scarb fmt`

Scarb projects can be formatted using the `scarb fmt` command.

If you're using the Cairo binaries directly, you can run `cairo-format` instead.

Many collaborative projects use `scarb fmt` to prevent arguments about which style to use when writing Cairo: everyone formats their code using the tool.

To format any Cairo project, enter the following inside the project directory:

```
```bash
scarb fmt
```
```

IDE Integration Using `cairo-language-server`

To help IDE integration, the Cairo community recommends using the

[`cairo-language-server`][cairo-language-server]<!-- ignore -->. This tool is a set of compiler-centric utilities that speaks the [Language Server Protocol][lsp]<!-- ignore -->, which is a specification for IDEs and programming languages to communicate with each other. Different clients can use `cairo-language-server`, such as [the Cairo extension for Visual Studio Code][vscode-cairo].

[lsp]: <http://langserver.org/>

[vscode-cairo]: <https://marketplace.visualstudio.com/items?itemName=starkware.cairo1>

Visit the `vscode-cairo` [page][vscode-cairo]<!-- ignore -->

to install it on VSCode. You will get abilities such as autocompletion, jump to definition, and inline errors.

[cairo-language-server]: <https://github.com/starkware-libs/cairo/tree/main/crates/cairo-lang-language-server>

> Note: If you have Scarb installed, it should work out of the box with the Cairo VSCode extension, without a manual installation of the language server.