

# Move Analyzer VS Code Extension

The Move Analyzer extension for Visual Studio Code provides language support features for the Move programming language. It enables syntax highlighting, code completion, and advanced features like definition linking and type checking.

You can install the Move extension from the Visual Studio Code Marketplace:

Alternative install methods include:

Use `Ctrl + P` or `⌘ + P` and type `ext install mysten.move`.

Use the command line:

The following extensions are included in the Move extension install:

The Move extension attempts to install the appropriate move-analyzer binary for your platform. If this doesn't work, or you prefer to install it manually, build it with Cargo:

By default, the Move extension expects to find the move-analyzer binary in `~/.sui/bin`. You can either copy the binary to this location, or configure the extension to use a different path.

The Move extension supports most Language Server Protocol features, as well as basic commands for building, testing, and tracing Move code.

The Move extension installs command palette commands for building, testing, and tracing Move code.

These commands find the `Move.toml` file for the open Move source file and open a terminal to run the appropriate `sui move` command.

To generate a trace, you must have a trace-capable `sui` binary. See [Debugger](#) for more information.

The [Move Syntax](#) extension provides syntax highlighting.

Hovering over identifiers shows type information, struct fields and attributes, and docstrings (if any) for the identifier. This works for all Move symbols including macros.

Hover over structs to see structure and definition.

Hover over functions for details and definition.

Hover over macros for their functionality.

The Move extension autocompletes upon a dot operator, displaying the available methods and fields for the type.

The Move extension also autocompletes after a `::` operator.

Finally, the Move extension provides "inlay hints," where the plugin automatically inserts the correct type after a variable declaration, unpack, function parameters, and other places.

The Move extension supports go-to-definition navigation for all Move symbols including types, functions, and macros, as long as the type was present when move-analyzer last built the file.

The extension also supports find-references for functions, macros, constants, and types.

## Install

You can install the Move extension from the Visual Studio Code Marketplace:

Alternative install methods include:

Use `Ctrl + P` or `⌘ + P` and type `ext install mysten.move`.

Use the command line:

The following extensions are included in the Move extension install:

The Move extension attempts to install the appropriate move-analyzer binary for your platform. If this doesn't work, or you prefer to install it manually, build it with Cargo:

By default, the Move extension expects to find the move-analyzer binary in `~/.sui/bin`. You can either copy the binary to this location, or configure the extension to use a different path.

The Move extension supports most Language Server Protocol features, as well as basic commands for building, testing, and tracing Move code.

The Move extension installs command palette commands for building, testing, and tracing Move code.

These commands find the `Move.toml` file for the open Move source file and open a terminal to run the appropriate `sui move` command.

To generate a trace, you must have a trace-capable `sui` binary. See [Debugger](#) for more information.

The [Move Syntax](#) extension provides syntax highlighting.

Hovering over identifiers shows type information, struct fields and attributes, and docstrings (if any) for the identifier. This works for all Move symbols including macros.

Hover over structs to see structure and definition.

Hover over functions for details and definition.

Hover over macros for their functionality.

The Move extension auto completes upon a dot operator, displaying the available methods and fields for the type.

The Move extension also auto completes after a `::` operator.

Finally, the Move extension provides "inlay hints," where the plugin automatically inserts the correct type after a variable declaration, unpack, function parameters, and other places.

The Move extension supports go-to-definition navigation for all Move symbols including types, functions, and macros, as long as the type was present when move-analyzer last built the file.

The extension also supports find-references for functions, macros, constants, and types.

## Features

The Move extension supports most Language Server Protocol features, as well as basic commands for building, testing, and tracing Move code.

The Move extension installs command palette commands for building, testing, and tracing Move code.

These commands find the `Move.toml` file for the open Move source file and open a terminal to run the appropriate `sui move` command.

To generate a trace, you must have a trace-capable `sui` binary. See [Debugger](#) for more information.

The [Move Syntax](#) extension provides syntax highlighting.

Hovering over identifiers shows type information, struct fields and attributes, and docstrings (if any) for the identifier. This works for all Move symbols including macros.

Hover over structs to see structure and definition.

Hover over functions for details and definition.

Hover over macros for their functionality.

The Move extension auto completes upon a dot operator, displaying the available methods and fields for the type.

The Move extension also auto completes after a `::` operator.

Finally, the Move extension provides "inlay hints," where the plugin automatically inserts the correct type after a variable declaration, unpack, function parameters, and other places.

The Move extension supports go-to-definition navigation for all Move symbols including types, functions, and macros, as long as the type was present when move-analyzer last built the file.

The extension also supports find-references for functions, macros, constants, and types.