

# Move Trace Debugger

The [Move Trace Debugger](#) extension for Visual Studio Code provides a familiar debugging interface for Move unit tests. You can step through code execution, track local variable values, and set line breakpoints to understand how your Move code executes.

You must have the [Move extension](#) installed to use the debugger. The Move extension includes the Move Trace Debugger so you should not need to install it separately. The install instructions are included for rare cases where an individual install might be necessary.

The Move Trace Debugger extension is available in the Visual Studio Code Marketplace. Search for Move Trace Debugger in the Extensions view, or press Ctrl + P or ⌘ + P and type ext install mysten.move-trace-debug .

Alternatively, run code --install-extension mysten.move-trace-debug to install the extension from the command line.

To use the debugger, you must also have the sui binary installed with the tracing feature flag enabled. The sui binaries in release tarballs, Homebrew, and Chocolatey have this feature enabled. See [Install Sui](#) for more information, including how to build from source.

Debugging a Move unit test is a two-step process:

## I. Generate execution traces

Open the command palette ( Shift + ⌘ + P on macOS, Ctrl + Shift + P on Windows/Linux).

Run the Move: Trace Move test execution command.

The extension displays a filter prompt. Either type a filter string to target specific tests or leave the field blank to run all tests and press Enter .

Find the generated traces in the traces directory.

## II. Start debugging

Open the Move file containing your test.

Select Run -> Start Debugging from the menu.

If the file has multiple tests, select the specific test from the dropdown menu.

Currently, the Move Trace Debugger supports basic forward debugging through test execution traces.

Support for reverse debugging and watch expressions is not currently available.

Move Trace Debugger supports standard debugging features like step over, step into, step out, continue, and stop. You can step through normal code and Move macros.

Move Trace Debugger supports displaying the values of primitive types, Move structs, and references.

At present, the debugger doesn't support setting watch points on variables.

## Install

You must have the [Move extension](#) installed to use the debugger. The Move extension includes the Move Trace Debugger so you should not need to install it separately. The install instructions are included for rare cases where an individual install might be necessary.

The Move Trace Debugger extension is available in the Visual Studio Code Marketplace. Search for Move Trace Debugger in the Extensions view, or press Ctrl + P or ⌘ + P and type ext install mysten.move-trace-debug .

Alternatively, run code --install-extension mysten.move-trace-debug to install the extension from the command line.

To use the debugger, you must also have the sui binary installed with the tracing feature flag enabled. The sui binaries in release tarballs, Homebrew, and Chocolatey have this feature enabled. See [Install Sui](#) for more information, including how to build from source.

Debugging a Move unit test is a two-step process:

#### I. Generate execution traces

Open the command palette ( Shift + ⌘ + P on macOS, Ctrl + Shift + P on Windows/Linux).

Run the Move: Trace Move test execution command.

The extension displays a filter prompt. Either type a filter string to target specific tests or leave the field blank to run all tests and press Enter .

Find the generated traces in the traces directory.

#### II. Start debugging

Open the Move file containing your test.

Select Run -> Start Debugging from the menu.

If the file has multiple tests, select the specific test from the dropdown menu.

Currently, the Move Trace Debugger supports basic forward debugging through test execution traces.

Support for reverse debugging and watch expressions is not currently available.

Move Trace Debugger supports standard debugging features like step over, step into, step out, continue, and stop. You can step through normal code and Move macros.

Move Trace Debugger supports displaying the values of primitive types, Move structs, and references.

At present, the debugger doesn't support setting watch points on variables.

## Usage

To use the debugger, you must also have the sui binary installed with the tracing feature flag enabled. The sui binaries in release tarballs, Homebrew, and Chocolatey have this feature enabled. See [Install Sui](#) for more information, including how to build from source.

Debugging a Move unit test is a two-step process:

#### I. Generate execution traces

Open the command palette ( Shift + ⌘ + P on macOS, Ctrl + Shift + P on Windows/Linux).

Run the Move: Trace Move test execution command.

The extension displays a filter prompt. Either type a filter string to target specific tests or leave the field blank to run all tests and press Enter .

Find the generated traces in the traces directory.

#### II. Start debugging

Open the Move file containing your test.

Select Run -> Start Debugging from the menu.

If the file has multiple tests, select the specific test from the dropdown menu.

Currently, the Move Trace Debugger supports basic forward debugging through test execution traces.

Support for reverse debugging and watch expressions is not currently available.

Move Trace Debugger supports standard debugging features like step over, step into, step out, continue, and stop. You can step through normal code and Move macros.

Move Trace Debugger supports displaying the values of primitive types, Move structs, and references.

At present, the debugger doesn't support setting watch points on variables.

## Features

Currently, the Move Trace Debugger supports basic forward debugging through test execution traces.

Support for reverse debugging and watch expressions is not currently available.

Move Trace Debugger supports standard debugging features like step over, step into, step out, continue, and stop. You can step through normal code and Move macros.

Move Trace Debugger supports displaying the values of primitive types, Move structs, and references.

At present, the debugger doesn't support setting watch points on variables.