Announcing our Fully Featured, Portable Solidity Debugger

Every one at Truffle is giddy right now.

Since Truffle's inception, it's been our mission to build outstanding development tools for the Ethereum community. The analogy we always used was one of history: Ethereum development practices are years behind the rest of the industry, and it's our job to build tools that modernize our craft. Well today, Truffle's come one step closer to that goal. We're happy to announce the release of our fully featured, fully portable Soliditydebugger anddebugging libraries .

What We've Built¶

If you've followed Truffle development, you're familiar with thetruffle debug command. This was the first incarnation of a "debugger" built into Truffle, though it wasn't much more than a code tracer. Our new debugger comes with everything you expect :

- Code stepping (over, into, out, next, instruction, etc.)
- Current code location, including the address of the running contract
- Breakpoints
- Watch expressions
- Variable inspection (stack, memoryand
- storage)
- Custom expression evaluation using Solidity variables

On top of that, our new debugger is fully portable, built to integrate with any Javascript project or editor, like Visual Studio Code. Lastly, we built it to work withany Ethereum client , so you could debug transactions against the main Ethereum network if you so desired.

How it Works¶

To debug your Solidity code, you first need to make a transaction. Ethereum is a world computer with a history of all execution, so we can pull all the information we need from past transactions. After obtaining the transaction hash, simply run the following to kick off your debugging experience:

truffle debug This will open up the debugging console which you can use to your heart's content!

The above user interface gives you access to the debugger features you need to step through your contracts' code, set breakpoints, evaluate expressions and inspect Solidity variables, providing you a better sense of what your code is doing. Under the hood, our debugger is interacting with your Ethereum client to gather all the data necessary to provide this information to you in a suitable manner. For developers reading this, the debugging library stores all this data in Redux, making the data easy to manage and making the library super portable.

Most of the work done by the debugger results in some type of mapping. The debugger gathers tons of disparate data, like the bytecode of your contracts, abstract representations of your code, instructions run during your transaction, etc., and combines them to produce information useful to the contract developer. Perhaps the debugger's most valuable property is variable inspection: Using all the data gathered, the debugger can determine which variables exist within your contract, which apply to each section of the code, and what their values are at any point in time. It can even show the values of complex data types, like structs with nested arrays. We're still filling out a few data types, but expect those in the next couple weeks.

If you plan on using thedebug commandtoday , we recommend usingUsing Truffle Develop and the Console . Truffle Develop comes with everything you need "baked in" to fully debug your contracts. Ganache andganache-cli will need to be updated, so stay tuned for that in the coming days.

Where to Next¶

The debugger we built unlocks tons of value for all Ethereum developers both new and advanced, but what you're seeing here is the tip of the iceberg of where this technology will go. The interface above is justone interface, meant to show off the debugger's capability. As mentioned, the libraries we built are super portable, so they'll end up making their way into all of the following:

- Integrated Development Environments (IDEs) like Visual Studio Code
- Ganache
- , including a full debugging UI
- Drizzle
- , for automatic variable change detection after a transaction
- Browser plugins so you can debug transactions on the fly within your application

- Ethereum provider libraries that let you debug your code anywhere, without a plugin

We encourage the Ethereum community to help build these tools. Our code is open source, and you can find all of it on Github .

How to Get It¶

If you already have Truffle installed, simply run the following to get you to the latest version:

npm uninstall -g truffle

npm install -g truffle If you're new to Truffle, you can try it out using:

npm install -g truffle You may have to usesudo with the above commands if you're running Linux.

More information about how to use Truffle can be found in our documentation , and feel free to reach out on our community GitHub Discussions channel , where hundreds of your fellow Trufflers congregate to answer your question.

A Quick Thank You¶

I'd like to personally thank the whole Truffle team for this effort, and particularly Nick D'Andrea for turning some initial research work into a fully portable, fully featured debugger. The engineering under the hood is amazing, I urge you to take a look . I'd also like to thank the community at large: It's a pleasure building the tools that make your lives easier, and your consistent feedback makes the whole endeavor worthwhile.

Thanks, -- Tim