

Stack Tracing with Truffle Test

Truffle Test can now provide stacktraces when your transactions revert! This is still a little experimental, and it requires `truffle test` to recompile all your contracts, so you'll have to enable it with `truffle test --stacktrace`. Here's what the result looks like:

`truffle test --stacktrace` The stacktrace continues seamlessly from the Javascript test into the Solidity contracts. It begins in the Javascript where the transaction was sent, continues through the contract that initially received the transaction, and finally ends in the contract where the revert occurred.

These Solidity stacktraces will be printed both for reverted transactions sent during tests and for reverted deployments made during tests with `Contract.new()`. Stacktraces are not currently available for calls or gas estimates. Note that stacktraces are a Truffle Contract feature, and will not be available if transactions are sent by other means.

What if we want a little more? Looking at the above example again, one of our stacktraces helpfully included a revert message, but the other did not. Wouldn't it be nice if it had one too? Let's try this again, but this time with `--stacktrace-extra`:

`--stacktrace-extra` There we go! The `--stacktrace-extra` option, in addition to turning on stacktraces, additionally compiles your contracts with Solidity's debug mode, so you get more information about just why the transaction reverted.

Of course, for more detailed analysis, you may want to use Truffle Debugger! But stacktraces in tests can give a good first view of why your transaction reverted.

These testing options are still a little experimental, so it's possible some things may not work with them. But, in most cases, they should provide a quick helpful view into why your test transaction reverted!

Thanks,

Harry Altman

--Debugger Engineer, Truffle Suite

[Click here for the release notes on GitHub](#)