# 538.https://stackoverflow.com/questions/70310087/how-do-i-resolve-this-hardhat-compilererror-stack-too-deep-when-compiling-inli

**T:**How do I resolve this Hardhat CompilerError? (Stack too deep when compiling inline assembly)

**Q:**I'm analyzing the Chainrunners smart contracts, so I went on Etherscan and copied the verified contract source code.  
  
When I tried to compile without solidity optimizer, I got this warning:  
  
thatguyintech@albert chainrunners % npx hardhat compileCompiling 5 files with 0.8.4Warning: Unused local variable. --> contracts/ChainRunnersBaseRenderer.sol:232:124: |232 | ... kenPalettes, uint8 numTokenLayers, string[NUM\_LAYERS] memory traitTypes) = getTokenData(\_dna); | ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off revert strings, or using libraries. --> contracts/ChainRunnersBaseRenderer.sol:48:1: |48 | contract ChainRunnersBaseRenderer is Ownable, ReentrancyGuard { | ^ (Relevant source part starts here and spans across multiple lines).  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So I tried to turn on the optimizer according to the Hardhat official documentation: https://hardhat.org/config/  
  
So here is what my Hardhat config hardhat.config.js looks like:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
/\*\* \* @type import('hardhat/config').HardhatUserConfig \*/module.exports = { solidity: { version: "0.8.4", settings: { optimizer: { enabled: true, runs: 2000, } } }};  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So now I am getting this hardhat CompilerError when I try to run npx hardhat compile:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
thatguyintech@albert chainrunners % npx hardhat compileCompiling 5 files with 0.8.4CompilerError: Stack too deep when compiling inline assembly: Variable value0 is 3 slot(s) too deep inside the stack.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Anyone know how I can resolve this? From a couple of google searches on hardhat-related threads, it seems like turning the optimizer on should be the fix to this issue, so I'm pretty confused.  
  
Here's an example I found on the OpenZeppelin forums that is not working for me: https://forum.openzeppelin.com/t/stack-to-deep-when-compiling-inline-assembly/11391/11

**C1:**I also just realized that this is not the actual Chainrunners nft contract -- it's the renderer contract. Now, I need to see how the nft contract uses the renderer...

2 **Answer**

**A1:**Ah it turns out there's a section in the Etherscan page that shows the exact solidity optimizer set. (h/t @alcuadadro)  
  
It looks like this:  
  
  
  
And so I copied that into my hardhat.config.js:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
/\*\* \* @type import('hardhat/config').HardhatUserConfig \*/module.exports = { solidity: { version: "0.8.4", settings: { optimizer: { enabled: true, runs: 2000, details: { yul: true, yulDetails: { stackAllocation: true, optimizerSteps: "dhfoDgvulfnTUtnIf" } } } }, },};  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
and that did the trick!  
  
no idea what the yul stuff is about though  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**yul: true and stackAllocation: true are just defaults when the optimizer is enabled. They're superfluous here but no harm in adding them.

**C2:**optimizerSteps: "dhfoDgvulfnTUtnIf" on the other hand is an advanced optimization. It disables most of the optimizer step sequence. The default sequence is quite a bit longer. I would not recommend it as a general solution to the "Stack too deep" error (it'll be solved in the compiler eventually) but it's quite interesting that disabling these optimizations actually helps in this case.

**C3:**thanks for the clarity! @cameel

**A2:**there is a size limit for contract to be compiled.  
  
Size cap of contract content is about 24577 bytes. Optimizer decreases the size. Default { enabled : false, runs : 200 }. runs can be set to up to 5000. this property helped you to compile the contract  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
In your case you did not need yul option. It helps you develop smart contracts with Yul and/or Yul+. https://docs.soliditylang.org/en/v0.8.17/yul.html