# 371.https://stackoverflow.com/questions/71368221/elrond-mandos-test-elrond-wasm-debugmandos-rs-pass-however-erdpy-contract-test

**T:**Elrond mandos test elrond\_wasm\_debug::mandos\_rs pass however erdpy contract test fail

**Q:**I'm writing test cases for my NFT smart contract (SC). When I check the state of the SC after creating my NFT I'm expecting to see a variable (next\_index\_to\_mint:u64, that's I increase by 1 every new NFT) to be updated.  
  
So I'm running the test using the command:  
  
$ erdpy contract testINFO:projects.core:run\_tests.project: /Users/<user>/sc\_nftINFO:myprocess:run\_process: ['/Users/<user>/elrondsdk/vmtools/mandos-test', '/Users/<user>/sc\_nft/mandos'], in folder: NoneCRITICAL:cli:External process error:Command line: ['/Users/<user>/elrondsdk/vmtools/mandos-test', '/Users/<user>/sc\_nft/mandos']Output: Scenario: buy\_nft.scen.json ... FAIL: wrong account storage for account "sc:nft-minter": for key 0x6e657874496e646578546f4d696e74 (str:nextIndexToMint): Want: "0x02". Have: ""Scenario: create\_nft.scen.json ... FAIL: wrong account storage for account "sc:nft-minter": for key 0x6e657874496e646578546f4d696e74 (str:nextIndexToMint): Want: "0x02". Have: ""Scenario: init.scen.json ... okDone. Passed: 1. Failed: 2. Skipped: 0.ERROR: some tests failed  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
However, when I'm running the test using elrond\_wasm\_debug::mandos\_rs function with the create\_nft.scen.json file, it passed.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
use elrond\_wasm\_debug::\*;fn world() -> BlockchainMock { let mut blockchain = BlockchainMock::new(); blockchain.set\_current\_dir\_from\_workspace(""); blockchain.register\_contract\_builder("file:output/test.wasm", nft\_auth\_card::ContractBuilder); blockchain}#[test]fn create\_nft() { elrond\_wasm\_debug::mandos\_rs("mandos/create\_nft.scen.json", world());}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
BTW, if you want to add this to the NFT SC example, that would be great in the tests/ folder.  
  
I tried to put an incorrect value, and it failed as expected. So my question is how could it be possible that it work using mandos elrond\_wasm debug but not erdpy ?  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
running 1 testthread 'create\_nft' panicked at 'bad storage value. Address: sc:nft-minter. Key: str:nextIndexToMint. Want: "0x04". Have: 0x02', /Users/<user>/elrondsdk/vendor-rust/registry/src/github.com-1ecc6299db9ec823/elrond-wasm-debug-0.28.0/src/mandos\_step/check\_state.rs:56:21  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here is the code (I use the default NFT SC example):  
  
const NFT\_INDEX: u64 = 0;fn create\_nft\_with\_attributes<T: TopEncode>(...) -> u64 { ... self.next\_index\_to\_mint().set\_if\_empty(&NFT\_INDEX); let next\_index\_to\_mint = self.next\_index\_to\_mint().get(); self.next\_index\_to\_mint().set(next\_index\_to\_mint+1); ...}#[storage\_mapper("nextIndexToMint")]fn next\_index\_to\_mint(&self) -> SingleValueMapper<u64>;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

1 **Answer**

**A1:**Short answer: most likely you haven't re-built your contract before testing it with erdpy.  
  
Long answer: currently there are two ways mandos tests are executed, as you've exemplified in your case:  
  
 ● Run tests directly from rust through mandos\_rs  
  
 ● Run tests through erdpy (which in turn uses mandos\_go)  
  
These two frameworks (mandos\_rs and mandos\_go) are functioning in different ways:  
  
 ● mandos\_rs: this framework is running on your rust code directly and it's testing it agains a mocked VM and mocked blockchain in the background. Therefore, it's not necessary to build your contract when using mandos\_rs.  
  
 ● mandos\_go: this framework is testing your compiled contract against a  
REAL VM with mocked blockchain in the background, so it's necessary to build your latest changes into a .wasm bytecode (e.g. erdpy contract build) before running the tests via mandos\_go, as this compiled file will be loaded by the VM like in a real use scenario.