目录

[1 Core: Code Explain 2](#_Toc18301)

[1.1 UniveV2ERC20.sol 2](#_Toc17631)

[1.2 UniswapV2Pair.sol 3](#_Toc2631)

[1.3 UniswapV2Factory 4](#_Toc15379)

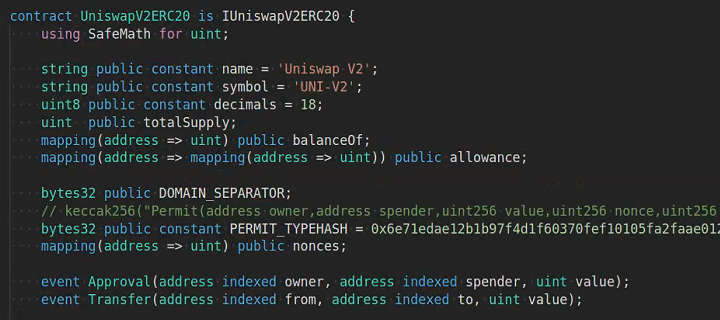
[2 Periphery: Code Explain 4](#_Toc12438)

[2.1 UniswapV2Router1 4](#_Toc17802)

[2.2 UniswapV2Migrator 6](#_Toc16038)

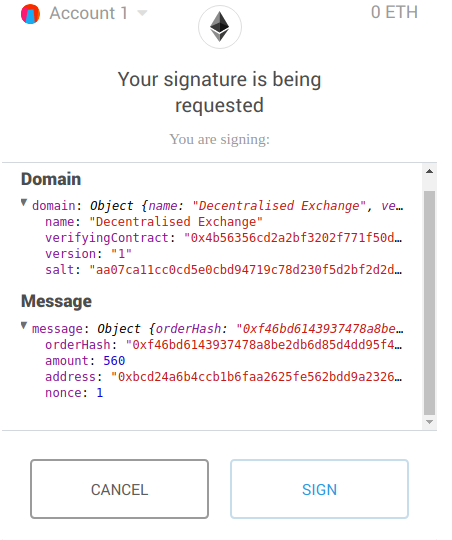
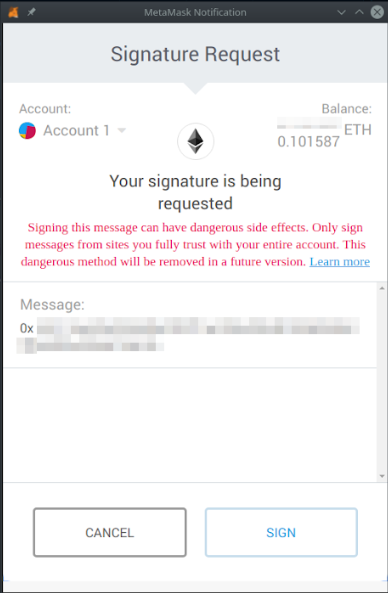
# 1 Core: Code Explain

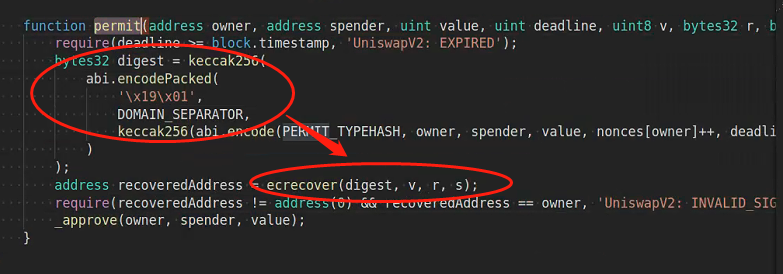
## 1.1 UniveV2ERC20.sol



- balanceOf: keep track of the univ2 liquidity balance

-eip712: before vs after

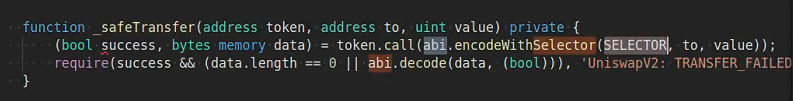




Recovers account that signed this data, and verify the address

## 1.2 UniswapV2Pair.sol





- solidity inline encoding:

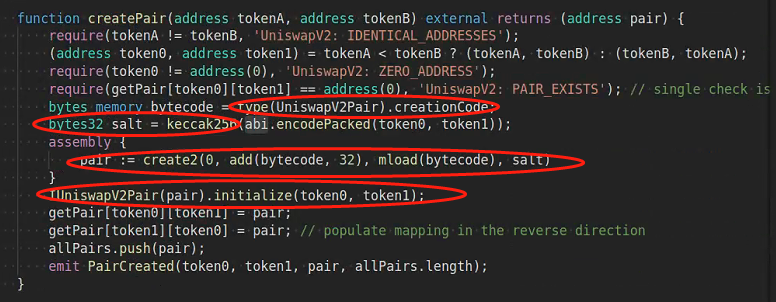
Token.call(abi.encodeWithSelector(SELECTOR, to, value) is exciting transfer function with to=address and uint256=value

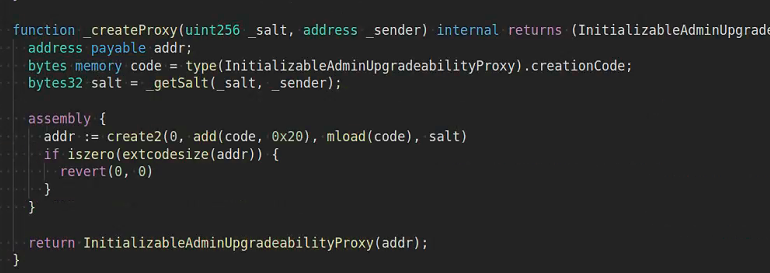


Mint function auto adjust reserves when new liquidity tokens are minted

So as to burn/ swap the reserve should be updated

## 1.3 UniswapV2Factory





The core parts of these 2 functions are the same

Salt

Memory

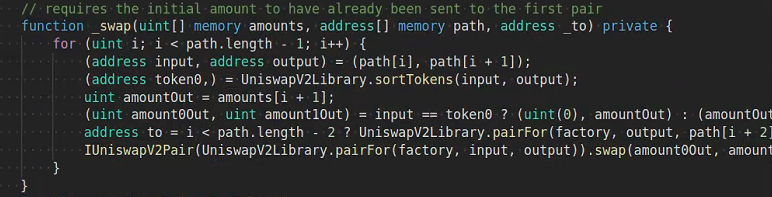
Create2

The only difference is the createPair is customized version of ProxyFactory

# 2 Periphery: Code Explain

## 2.1 UniswapV2Router1

Example:

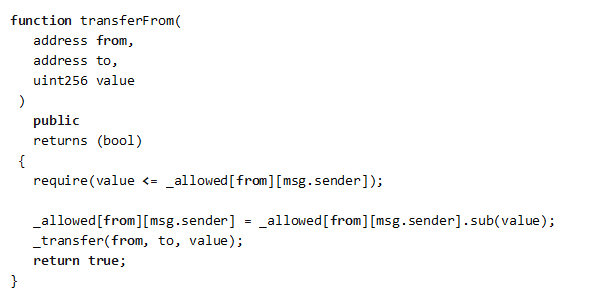


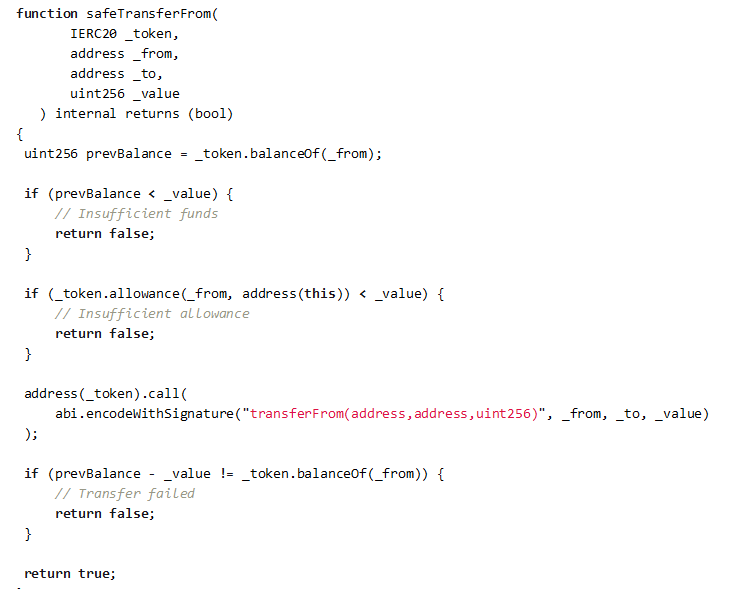
How to use functions from other contracts:

1 add interface

2 instantiate interface at actual address

3 call abstract function



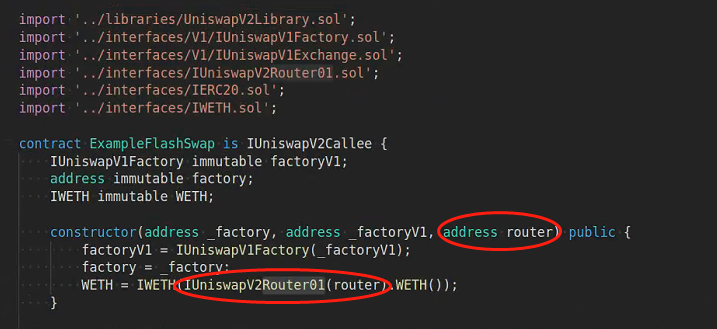


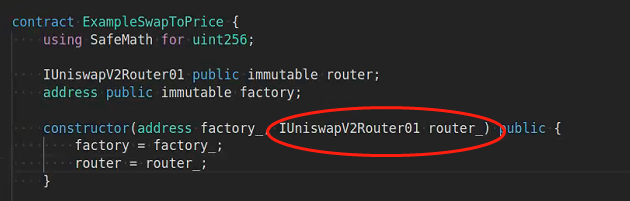
* Check if the value to be transferred is lower or equal to the account balance.
* Check if the value to be transferred is lower or equal to the allowance of the account which is going to perform the transfer.
* Check if account balance after the transfer is equal to previous balance minus the value transferred

2.2 how to use Router contract

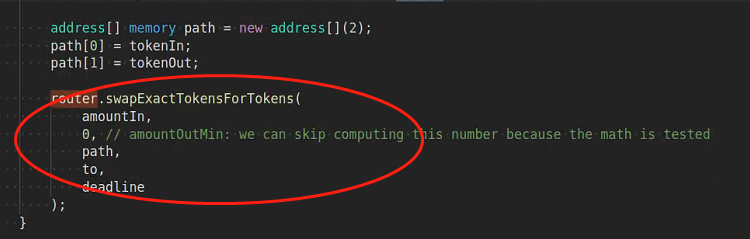
ExampleSwapToPrice

Step1: initiate router

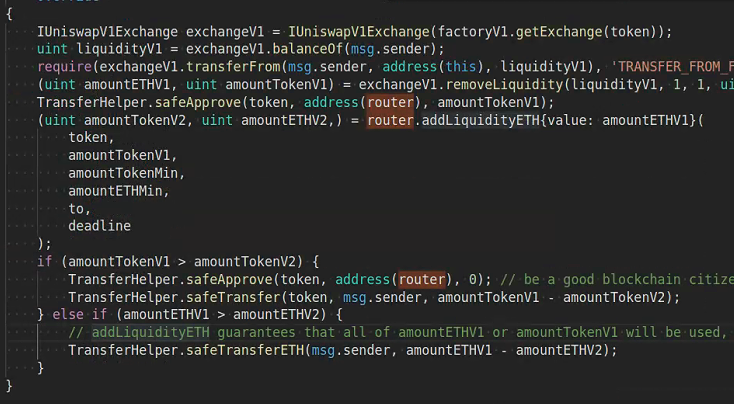




Step2 call function



## 2.2 UniswapV2Migrator



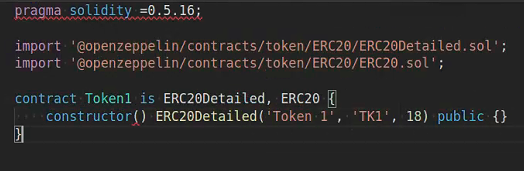
How to migrate

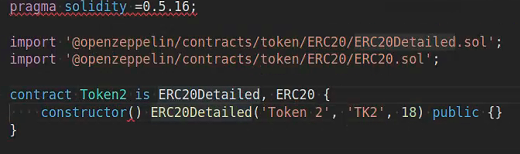
Use old exchangeV1 to remove Liquidity

And then use router to add the liquidity back

# 3 How to Fork Uniswap

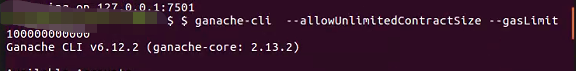
## Step1:create 2 test tokens under the core directory



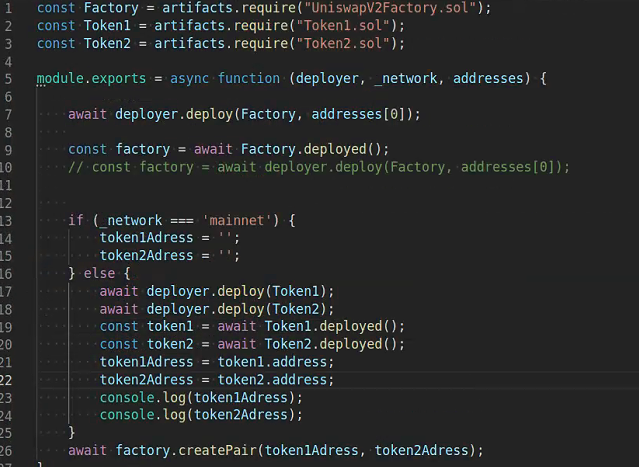


## Step2: deploy create trading pair contract

Start ganache:

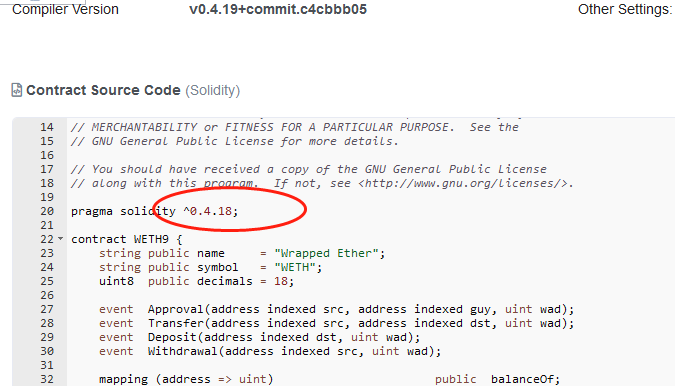


Truffle migrate



Step 3: write WETH in periphery

<https://etherscan.io/address/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2#code>



Change it to solidity 0.6

Step 4: deploy router



## Walk through test cases

Concept: approve

Why need approve?

Let's assume we have user **A** and user **B**. **A** has 1000 tokens and want to give permission to **B** to spend 100 of them.

* **A** will call approve(address(B), 100)
* **B** will check how many tokens **A** gave him permission to use by calling allowance(address(A), address(B))
* **B** will send to his account these tokens by calling transferFrom(address(A), address(B), 100)

# 3 Rif-marketplace NFT code walk through

Concepts:

1. Plausble: pause the whole contract
2. Payable: use to receive eth
3. How to do Whitelist

# Aave how to write scalable smart contracts