目录

[Summary: 2](#_Toc65844747)

[Smart Contract description: 3](#_Toc65844748)

[When the contract is deployed, the deployed date/time will be saved at “\_deployedDateTime”. 4](#_Toc65844749)

[Create “deployedDateTime” Function to check the contract deployed date/time 4](#_Toc65844750)

[Create function to check if the contract is 30 days old 5](#_Toc65844751)

[Update ERC20 contract and return balance as zero for all address when the contract 30 days old 5](#_Toc65844752)

[Following functions are also disabled if the contract is 30 days old 6](#_Toc65844753)

[Pause contract to stop all transactions during the upgrade or anything is wrong 7](#_Toc65844754)

[Test Case:  ECR20 working well if age less than 30 days 7](#_Toc65844755)

# Summary:

The smart contract design as specified at this document is based on following requirement from Nick Friedland:

*Design an ERC20 compliant token where the balances reset every time you upgrade it.*

*If you own this stablecoin in January, and there is always a smart contract upgrade at the end of*

*each month, then in February, you should not own those stablecoins anymore. Your token*

*should have all ERC20 interfaces and events. You can use whatever framework or version of*

*solidity that you would like, but you cannot use a framework for handling the creation of the*

*proxy contracts. It would be great to have a few tests to demonstrate functionality without having*

*to manually run through things.*

# Smart Contract description:

This is an ERC20 tokens and it implement all ERC20 standard interfaces and events with some additional functions based on requirement as specified at the summary section. This smart contract is built and verified under “Remix”, all the screenshots at this document are captured under Remix.

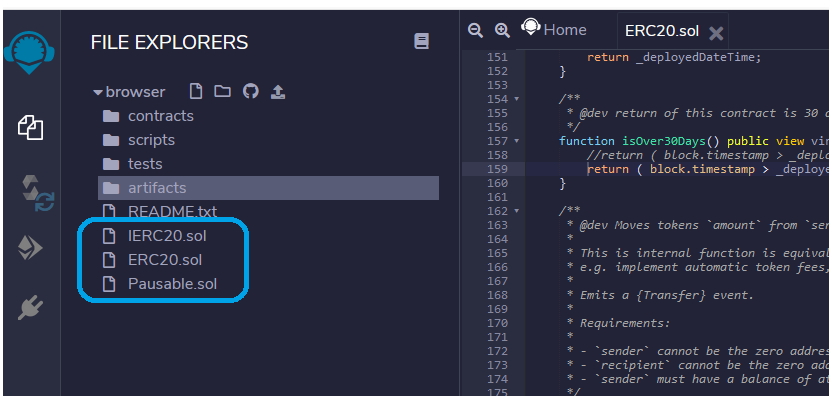
Before I start to work with this contract based on my understanding with this requirement, I think there is a challenger. Following describer the change:

Based on requirement, the balance need to be reset after 30 days. In order to reset all balance against all address, normally we need walk all address and reset their balance as specified in a mapping “balance” field. It is pretty expensive as we need another array to keep all users since there is no way to get all addresses from the mapping balance. And also we need a loop to walk all address and reset the balance as zero. And that may cause “run out of gas” issue if there is big number of address.

In order to fix above issue, I created a function to check the contract age. The contract will always return the balance as zero when the contract is 30 days old. And also some of the functions are also disabled, such as: Transfer, Approve etc. That also means all transaction will be reverted after 30 days.

There are 3 Solidify files:

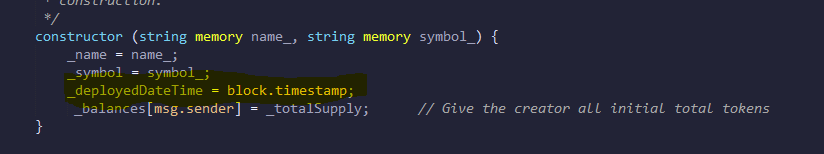
* IERC20.sol – This contract define all ERC20 interfaces and events
* Pausable.sol – Define Pause/Unpause relates functions and Modifier
* ERC20.sol – This is the ERC20 smart contract (Fixed total supply with 6 decimals). Above 2 Solidities will be imported to this contract



This document doesn’t provide description for those ERC20 standard features. Only those additional functions are highlighted.

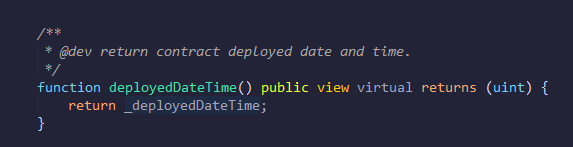
## When the contract is deployed, the deployed date/time will be saved at “\_deployedDateTime”.

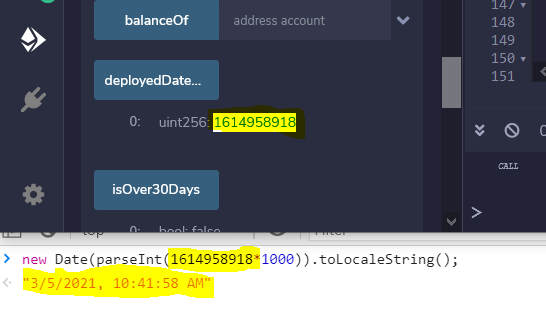
The contract deployed date/time will be used to calculate if the contract is reached 30 days.



## Create “deployedDateTime” Function to check the contract deployed date/time

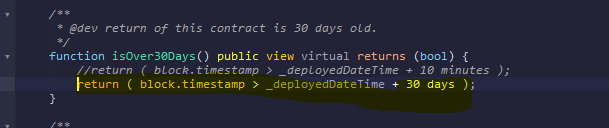
This function returns the contract deployed date/time. Such as the “deployedDateTime” returns “1614958918” as of my flowing testing. This is a UNIX date/time and it can be converted to readable date/time using Javascript which is “"3/5/2021, 10:41:58 AM"”. Please check more at below screenshot





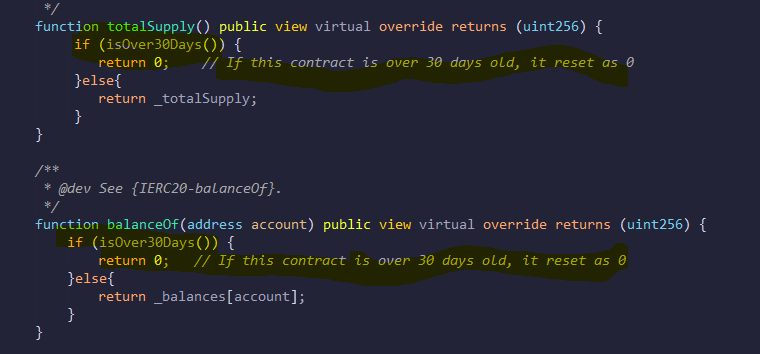
## Create function to check if the contract is 30 days old

This function compare current date/time with this contract initial deployed date/time, it returns true if the result is over 30 days old



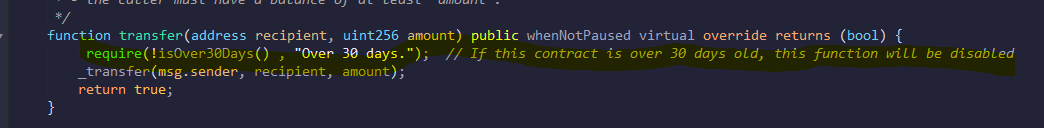
## Update ERC20 contract and return balance as zero for all address when the contract 30 days old

* Update following functions and returns zero balance for ALL address when this contract is over 30 days old
  + - totalSupply - Return zero as total supply
    - balanceOf – Return zero for ALL users
    - Allowance – Return zero for all allowance balance addresss

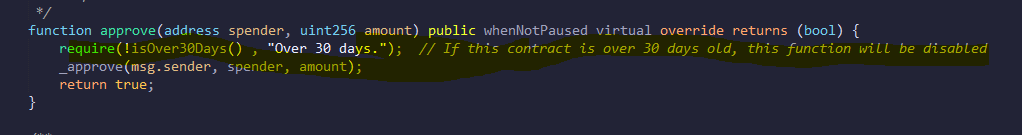


## Following functions are also disabled if the contract is 30 days old

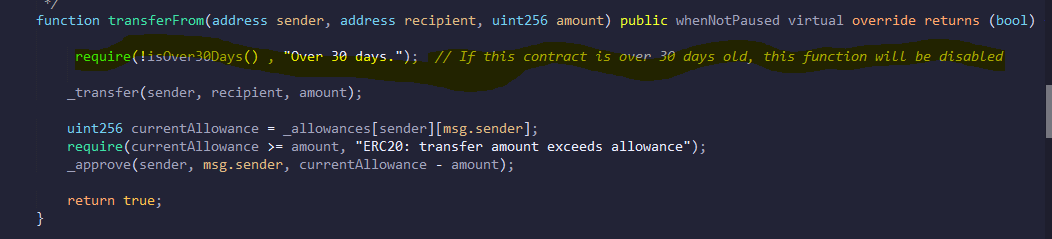
* + Transfer – All transfer transaction will be reverted after 30 days



* + Approve - All approval transaction will be reverted after 30 days



* + transferFrom – All transferFrom transaction will be reverted after 30 days

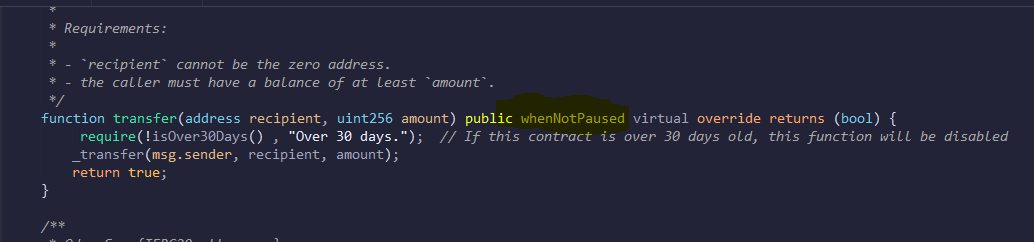


## Pause contract to stop all transactions during the upgrade or anything is wrong

Just in case if anything wrong, this contract can be paused in order to minimum the loss (Only owner can pause the contract) . After the contract is paused, all transactions will be reverted. We can also pause the contract during the upgrade.

The pause logic is defined as “Pausable.sol” files and this file is imported to the ERC20 contract. Only the contract owner has the ability to pause the contract.

Such as at following “transfer” function, it use “whenNotPaused” modifier. It will disable the transfer transaction if this contract is paused by owner



# Test Case:  ECR20 working well if age less than 30 days

|  |  |
| --- | --- |
| **Test Preparation** | * Update “isOver30Days” function and set the age as a small number (such as 10 minutes” at my testing so that we don’t need wait 30 days to test this case)      * Select “Javascript VM” from the environment list * Switch to the first address under the account list. And this address will be used to deploy the contract and own all initial supply tokens. And this address is also the owner of this contract |
| **Test Execution** | * Deploy the contract   Specify any token name and symbol (such as “Richard”, RICH” at following sample), click “Deploy” to deploy the contract     * Before the 30 days (It is 10 minutes at my case)   + Check Balance against the owner address, and it should return balance successfully   + Check total supply and it should return all total supply tokens   + Check “isOver30Days” and it should returns “False”   + Transfer 1000 tokens from owner address to another address, the transfer should be executed successfully        * After 30 days (10 minutes at this case)   + Check Balance against the any address, and it always should return zero   + Check total supply and it should return zero   + Check “isOver30Days” and it should returns “True”   + Transfer 1000 tokens from owner address to another address, the transfer should be reverted |
| **Expect Result** | * Before the contract age is less than 30 days, the all ERC20 functions work as expected * After the contract age is passing 30 days, all ERC20 transaction is disabled. And balance (include total supply) for ALL address returns zero |
| **Actual Result** |  |
| **Others** | * Tester: * Test Date: * Comment: |