Corion - CorionX Token

Technical documentation

Table of content

Code folder structure

The actual source code is to be found under the following git repo:
https://github.com/CORIONplatform/CorionX

Branching is kept simple at the moment, the master branch contains the major live version of the token and dependencies on the live network as dev branch contains elements under development.
The structure of the repo is the following:

- contracts : contain the solidity contracts
- deploy: contains deployment specific items
- docs: contains some minimal rather developers specific documentation
- migrations: standard library from Truffle for migration scripts
- misc: contains at the moment only the ABI file
- services: contains code to realize the initial token allocation
- test : is a standard library from truffle, containing unit tests in mocha

Contracts

The following important solidity contracts were used during the development, most of them are standard OpenZeppelin implementations:

File	Description
ERC20Basic.sol	Simpler version of ERC20 interface
ERC20.sol	ERC20 interface
BasicToken.sol	Basic version of StandardToken, with no allowances.
BurnableToken.sol	Token that can be irreversibly burned (destroyed)
Ownable.sol	The Ownable contract has an owner address, and provides basic authorization control
StandardToken.sol	Implementation of the basic standard token
CorionXToken.sol	ERC20 token for the CorionX Token
SafeMath.sol	Math operations with safety checks that throw on error

Migrations.sol	standard truffle migration library

Stages

The token has been deployed and tested on all active Ethereum networks and finally on the live one. The latest addresses which used for the testing for the different stages are the followings:

- TEST1: 0x9b46BE2116F52f63B81bd98961efCDb4B0397540
- TEST2: 0x53cb49375f1A3f1331Cdf87054D73BBA08215634
- ROPSTEN: 0x18921a0F96805AbAEF9ce33ee0fD653f15f88B7d
  (CONTRACT: 0xe72f01A9d819C28C9b0A7c9cD6242224E5f73169)
- LIVE: 0x089aad9e9106624023ea15363dae8abf6f8b7b0c
  (CONTRACT: 0x26a604dffe3ddab3bee816097f81d3c4a2a4cf97)

ABI documentation

Source code

Sendable methods

The method thats need to be mined and costs some ether

approve

Approve an address to spent some token from owned tokens.

Parameters:
- _spender: The address who can spend the tokens
- _value: The amount of the spendable tokens.

decreaseApproval

Decrease the approved amount of spendable token for the specified address.

Parameters:
- _spender: The address who can spend the tokens
- _subtractedValue: The amount of the change of spendable tokens.

increaseApproval

Increase the approved amount of spendable token for the specified address.

Parameters:
- _spender: The address who can spend the tokens

- _addedValue: The amount of the change of spendable tokens.

transfer

Transfer the specified amount of token to other address.

Parameters:
- _to: The receiver address
- _value: The amount of token

transferFrom

Transfer the specified amount of token to other address from the specified address.
The method caller need to have approvement from the sender address and the amount of the token!

Parameters:
- _from: The sender address.
- _to: The receiver address
- _value: The amount of token

Callable methods

The method thats don't need to be mined. It is doesn't cost any more ether.

allowance

Check the allowed amount of the token, which can be send from the spender in the name of the owner.

Parameters:
- _owner: The owner address, who have the token amount
- _spender: The spender address who can handle the owner token

balanceOf

Check the owned amount of the token from owner address.

Parameters:
- _owner: The owner address, who have the token amount

Token properties

name

The fully qualified name of the token.

symbol

The symbol of the token

decimals

The maximum decimal of the token.

INITIAL_SUPPLY

The initial amount of the tokens. First owner of all token is the creator of the smart contract.

owner

The creator of the smart contract.

totalSupply

The actual amount of the tokens in the smart contract.