Dai - Detailed Documentation

The Dai Token Contract * Contract Name: * dai.sol * Type/Category: * DSS —> Dai Module * Associated MCD System Diagram * Contract Source * Etherscan *

1. Introduction (Summary)

TheDai contract is the user-facing ERC20 token contract maintaining the accounting for external Dai balances. Most functions are standard for a token with changing supply, but it also notably features the ability to issue approvals for transfers based on signed messages.

?

1. Contract Details

DAI (Glossary)

Key Functionalities (as defined in the smart contract)

Mint - Mint to an address

Burn - Burn at an address

Push - Transfer

Pull - Transfer From

Move - Transfer From

Approve - Allow pulls and moves

Permit - Approve by signature

Other

name - Dai Stablecoin

symbol - DAI

version - 1

decimals - 18

totalSupply - Total DAI Supply

balanceOf(usr: address) - User balance

allowance(src: address, dst: address) - Approvals

nonces(usr: address) - Permit nonce

Units

- wad
-
    - fixed point decimal with 18 decimals (for basic quantities, e.g. balances).
-

1. Key Mechanisms & Concepts

For the most part,dai.sol functions as a typical ERC20 token. These tokens have been already beenheavily documented here and it is recommended to read through that documentation for the core functions of an ERC20 token.

Differences From ERC20:

1. transferFrom
2. in the DAI contract works in a slightly different form than the generictransferFrom
3. function. The DAI contract allows for "unlimited approval". Should the user approve an address for the maximum uint256 value, then that address will have unlimited approval until told otherwise.
4. push
5. ,pull

6. &move
7. are aliases fortransferFrom
8. calls in the form oftransferFrom(msg.sender, usr, amount)
9. ,transferFrom(usr, msg.sender, amount)
10. &transferFrom(src, dst, amount)
11. .
12. permit
13. is a signature-based approval function. This allows for an end-user to sign a message which can then be relayed by another party to submit their approval. This can be useful for applications in which the end-user does not need to holdETH
14. .
15.
    - In order to use this functionality, a user's address must sign a message with theholder
16.
    - ,spender
17.
    - ,nonce
18.
    - ,expiry
19.
    - and theallowed
20.
    - amount. This can then be submitted toPermit()
21.
    - to update the user's approval.
22. *
23.

1. Gotchas (Potential Source of User Error)

Unlimited allowance is a relatively uncommon practice (though becoming more common). This could be something used to trick a user by a malicious contract into giving access to all their DAI. This is concerning in upgradeable contracts where the contract may appear innocent until upgraded to a malicious contract.

DAI is also susceptible to the knownERC20 race condition , but should not normally be an issue with unlimited approval. We recommend any users using theapproval for a specific amount be aware of this particular issue and use caution when authorizing other contracts to perform transfers on their behalf.

There is a slight deviation intransferFrom functionality: If thesrc == msg.sender the function does not requireapproval first and treats it as a normaltransfer from themsg.sender to thedst .

Built-in meta-transaction functionality of Dai

The Dai token provides offchain approval, which means that as an owner of an ETH address, you can sign a permission (using the permit() function) which basically grants allowance to another ETH address. The ETH address that you provide permission to can then take care of the execution of the transfer but has an allowance.

1. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

2. N/a

3.

Previous Dai Module Next Core Module Last updated4 years ago On this page *1. Introduction (Summary) * 2. Contract Details * DAI (Glossary) * 3. Key Mechanisms & Concepts * 4. Gotchas (Potential Source of User Error) * 5. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

Export as PDF