Dai Module

The DAI token contract and all of the adapters DaiJoin adapters. * Module Name: * DAI Module * Type/Category: Proxy —> * Dai.sol and DaiJoin.sol * Associated MCD System Diagram * Contract Sources: * *dai * * daiJoin * * *

1. Introduction (Summary)

The origin of DAI was designed to represent any token that the core system considers equal in value to its internal debt unit. Thus, theDAI Module contains the DAI token contract and all of the adapters DaiJoin adapters.

1. Module Details

Glossary (DAI)

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

Glossary (Join)

- vat
-
  - storage of the Vat’s address
- ilk
-
  - id of the Ilk for which aGemJoin
- is created for
- gem
-
  - the address of theilk
- for transferring
- dai
-
  - the address of thedai
- token
- one
-
  - a 10^27 uint used for math inDaiJoin
-

Core Module Components Documentation

1. Dai - Detailed Documentation
2. DaiJoin Documentation
3. (referenced in Join - Detailed Documentation)
4.

1. Key Mechanism and Concepts

Why are these components important to the Multi-Collateral Dai (MCD) System?

TheDai contract is the user facing ERC20 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.

Join consists of three smart contracts, one of which is the DaiJoin contract. Each join contract is created specifically to allow the given token type to be joined to the vat. Because of this, each join contract has slightly different logic to account for the different types of tokens within the system. The DaiJoin contract allows users to withdraw their Dai from the system into a standard ERC20 token.

1. Gotchas (Potential sources of user error)

2. DAI

3. is also susceptible to the knownERC20 race condition
4. , but should not normally be an issue with unlimited approval. We recommend any users using theapproval
5. for a specific amount be aware of this particular issue and use caution when authorizing other contracts to perform transfers on their behalf.
6. There are limited sources of user error in thejoin
7. contracts system due to the limited functionality of the system. Barring a contract bug, should a user call join by accident they could always get their tokens back through the corresponding exit call on the given join contract. The main issue to be aware of here would be a well-executed phishing attack. As the system evolves and potentially more join contracts are created, or more user interfaces are made, there is the potential for a user to have their funds stolen by a malicious join contract which does not actually send tokens to the vat, but instead to some other contract or wallet.
8.

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

There could potentially be a vat upgrade that would require new join contracts to be created.

If agem contract were to go through a token upgrade or have the tokens frozen while a user's collateral was in the system, there could potentially be a scenario in which the users were unable to redeem their collateral after the freeze or upgrade was finished. This seems to be a small risk though because it would seem likely that the token going through this upgrade would want to work alongside the maker community to be sure this was not an issue.

Previous Maker Protocol 101 Next Dai - Detailed Documentation Last updated4 years ago On this page *1. Introduction (Summary) * 2. Module Details * Glossary (DAI) * Glossary (Join) * Core Module Components Documentation * 3. Key Mechanism and Concepts * 4. Gotchas (Potential sources of user error) * 5. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

Export as PDF