Pot - Detailed Documentation

The Dai Savings Rate * Contract Name: * pot.sol * Type/Category: * DSS —> Rates Module * Associated MCD System Diagram * Contract Source * Etherscan *

1. Introduction (Summary)

The Pot is the core of theDai Savings Rate . It allows users to depositdai and activate the Dai Savings Rate and earning savings on theirdai . The DSR is set by Maker Governance, and will typically be less than the base stability fee to remain sustainable. The purpose of Pot is to offer another incentive for holding Dai.

?

1. Contract Details

Math

- mul(uint, uint)
- ,rmul(uint, uint)
- ,add(uint, uint)
- &sub(uint, uint)
-
    - will revert on overflow or underflow
- rpow(uint x, uint n, uint base)
- , used for exponentiation indrip
- , is a fixed-point arithmetic function that raisesx
- to the powern
- . It is implemented in assembly as a repeated squaring algorithm.x
- (and the result) are to be interpreted as fixed-point integers with scaling factorbase
- . For example, ifbase == 100
- , this specifies two decimal digits of precision and the normal decimal value 2.1 would be represented as 210;rpow(210, 2, 100)
- should return 441 (the two-decimal digit fixed-point representation of 2.1^2 = 4.41). In the current implementation, 10^27 is passed forbase
- , makingx
- and therpow
- result both of typeray
- in standard MCD fixed-point terminology.rpow
- 's formal invariants include "no overflow" as well as constraints on gas usage.
-

Auth

- wards
- are allowed to call protected functions (Administration)
-

Storage

- pie
-
    - stores the address'Pot
- balance.
- Pie
-
    - stores the total balance in thePot
- .
- dsr
-
    - thedai savings rate
- . It starts as1
- (ONE = 10^27
-), but can be updated by governance.
- chi
-
    - the rate accumulator. This is the always increasing value which decides how muchdai
-
    - given whendrip()

- is called.
- vat
  - an address that conforms to aVatLike
- interface. It is set during the constructor andcannot be changed
- .
- vow
  - an address that conforms to aVowLike
- interface. Not set in constructor. Must be set by governance.
- rho
  - the last time that drip is called.

The values ofdsr andvow can be changed by an authorized address in the contract (i.e. Maker Governance). The values ofchi ,pie ,Pie , andrho are updated internally in the contract and cannot be changed manually.

1. Key Mechanisms & Concepts

drip()

- Calculates the most recentchi
- and pullsdai
- from thevow
- (by increasing theVow
- 'sSin
-).
- A user should always make sure that this has been called before calling theexit()
- function.
- drip has to be called before a userjoin
- s and it is in their interest to call it again before theyexit
- , but there isn't a set rule for how often drip is called.

join(uint wad)

- uint wad
- this parameter is based on the amount of dai (sincewad
- =dai
- /chi
-) that you want tojoin
- to thepot
- . Thewad * chi
- must be present in thevat
- and owned by themsg.sender
- .
- themsg.sender
- 'spie
- amount is updated to include thewad
- .
- the totalPie
- amount is also updated to include thewad
- .

exit(uint wad)

- exit()
- essentially functions as the exact opposite ofjoin()
- .
- uint wad
- this parameter is based on the amount of dai that you want toexit
- thepot
- . Thewad * chi
- must be present in thevat
- and owned by thepot
- and must be less thanmsg.sender
- 'spie

- balance.
- Themsg.senders
- pie
- amount is updated by subtracting thewad
- .
- The totalPie
- amount is also updated by subtracting thewad
- .
-

Administration

Various file function signatures for administeringPot :

- Setting new dsr (file(bytes32, uint256)
-)
- Setting new vow (file(bytes32, address)
-)
-

Usage

The primary usage will be foraddresses to store theirdai in thepot to accumulate interest over time

1. Gotchas / Integration Concerns

2. Thedsr

3. is set (globally) through the governance system. It can be set to any number > 0%. This includes the possibility of it being set to a number that would cause the DSR to accumulate faster than the collective Stability Fees, thereby accruing system debt and eventually causing MKR to be minted.
4. Ifdrip()
5. has not been called recently before an address callsexit()
6. they will not get the full amount they have earned over the time of their deposit.
7. If a user wants tojoin
8. orexit
9. 1 DAI into/from the Pot, they should send awad
10. = to1 / chi
11. as the amount moved from their balance will be1 * chi
12. (for an example of this, seeDSS-Proxy-Actions
13.

1. Failure Modes and Impact

Coding Error

A bug in thePot could lead to locking ofdai if theexit() function or the underlyingvat.suck() orvat.move() functions were to have bugs.

Governance

Thedsr rate initially can be set through the Chief. Governance will be able to change the DSR based on the rules that the DS-Chief employs (which would include a Pause for actions).

One serious risk is if governance chooses to set thedsr to an extremely high rate, this could cause the system's fees to be far too high. Furthermore, if governance allows thedsr to (significantly) exceed the system fees, it would cause debt to accrue and increase the Flop auctions.

Previous Rates Module Next Jug - Detailed Documentation Last updated4 years ago On this page *1. Introduction (Summary) * 2. Contract Details * 3. Key Mechanisms & Concepts * Usage * 4. Gotchas / Integration Concerns * 5. Failure Modes and Impact

Export as PDF