

Dai Savings Rate

Alias: DSRParameter Name: dsr Containing Contract: MCD_POT Etherscan Link:

[0x197E90f9FAD81970bA7976f33CbD77088E5D7cf7](https://etherscan.io/address/0x197E90f9FAD81970bA7976f33CbD77088E5D7cf7) Scope: SystemTechnical Docs: [Pot Detailed Documentation](#)

Description

The Dai Savings Rate parameter allows Maker Governance to adjust the interest rate paid to DAI holders who have deposited DAI into the Dai Savings Rate contract (MCD_POT). All DAI Holders are permitted to deposit into the contract to receive the DSR. DAI is never locked in the Dai Savings Rate Contract, it is always accessible to the user that deposited it.

The Dai Savings Rate is usually expressed as an APY percentage and will increase balances when set to any positive value.

Example

Dai Savings Rate = 1%

1. User A deposits 100 DAI into the Dai Savings Rate Contract.
2. User A can withdraw 101 DAI after 1 year.
- 3.

Purpose

The Dai Savings Rate parameter enables Maker Governance to increase or decrease the incentive for DAI holders to deposit DAI within the Dai Savings Rate contract.

Trade-offs

Increasing the Dai Savings Rate increases DAI demand at the cost of decreased DAI liquidity and increased expenses for the Maker Protocol.

Decreasing the Dai Savings Rate will decrease the expenses of the Maker Protocol but may simultaneously reduce DAI demand and cause users to move DAI out of the Dai Savings Rate contract, thus increasing market supply.

Enhanced Dai Savings Rate

The [Enhanced Dai Savings Rate \(EDSR\)](#) is part of the [Maker Endgame Plan](#) and temporarily increases the DSR through a multiplier when the total Dai in the Dai Savings Rate contract is low relative to the total Dai supply. The EDSR multiplier decreases over time as DSR utilization increases. The EDSR is a non-increasing parameter i.e. even if DSR utilization goes down, the EDSR multiplier cannot increase with time. The EDSR multiplier tier is chosen according to the following table

Tier DSR Utilization DSR Multiplier
1 0 - 35% 1.75
2 35-50% 1.3
The EDSR is initialized in Tier 1. If the DSR utilization meets the threshold for a higher tier for a 24-hour period, the multiplier is adjusted manually through the next executive vote.

If the EDSR multiplier results in an effective EDSR above 5%, the effective EDSR remains capped at 5%. Maker Governance may disable the EDSR at any time through an executive vote or if the DSR utilization exceeds 50% for a 24-hour period.

Changes

Adjusting the Dai Savings Rate is a manual process that requires an executive vote. Changes to the Dai Savings Rate are subject to the [GSM Pause Delay](#).

In essence, a higher Dai Savings Rate results in upward pressure on Dai price, and a lower Dai Savings Rate will reduce this pressure.

Why increase this parameter?

The primary reason for increasing the Dai Savings Rate is to increase DAI demand. This increase in demand is achieved by encouraging DAI holders to deposit DAI to the Dai Savings Rate contract — thus removing it from the market, which should create upwards pressure on the dollar peg. Therefore, raising the Dai Savings Rate should be considered if DAI is trading below the dollar peg.

Increasing the Dai Savings Rate should make DAI a more attractive asset for holders than other stablecoin assets, potentially increasing DAI's market share in comparison to these assets. It may also incentivize additional integrations with DAI and the Maker Protocol in other dApps.

Why decrease this parameter?

The primary reason to decrease the Dai Savings Rate is to reduce DAI demand. Reducing the Dai Savings Rate should result in less DAI deposited in the Dai Savings Rate contract and cause that Dai to re-enter the market. In addition, it may be

advisable to decrease the Dai Savings Rate if DAI trades above the dollar peg to reduce upwards price pressure.

Maker Governance may also wish to decrease the Dai Savings Rate when The Maker Protocol experiences a drop in revenue. If Maker Governance does not reduce the Dai Savings Rate when income falls, the Maker Protocol could encounter negative cash flows. Negative cash flow will reduce the System Surplus Buffer over time, and if the buffer is depleted, the Protocol will mint and sell MKR to cover the shortfall.

Considerations

When DAI holders deposit DAI to the Dai Savings Rate contract, interest is paid from accrued stability fees. Therefore, increasing the Dai Savings Rate will cause the [System Surplus Buffer](#) to fill more slowly and reduce the amount of Dai available for Surplus Auctions. If the Dai Savings Rate is set too high, the Maker Protocol could have negative cash flow and eventually need to print MKR.

Other parameters and mechanisms that are dependent on the DSR, such as crypto-collateralized stability fees, are affected by the EDSR. This applies to all Vault Types with the exception of ETH-A, ETH-B, and ETH-C.

Balances within the Dai Savings Rate contract will not be updated unless the `drop` function is called; this may be done permissionlessly.

DAI deposited to the Dai Savings Rate contract is not held as an ERC-20 token balance within the contract. The balance of the contract is recorded in `MCD_VAT` and can be queried at any time by inputting the address of the `MCD_POT` contract (0x197E90f9FAD81970bA7976f33CbD77088E5D7cf7) into the `Dai` function of `MCD_VAT` (0x35D1b3F3D7966A1DFe207aa4514C12a259A0492B).

A negative Dai Savings Rate will not work as intended as balances within the Dai Savings Rate contract cannot decrease.

Any changes to the Dai Savings Rate may affect PSM usage behavior. For example, increased upward pressure on the dollar peg caused by the Dai Savings Rate may require increased PSM usage to balance.

In the event of an [Emergency Shutdown](#), the Dai Savings Rate will be set to 0% to prevent the total debt of the Maker Protocol from increasing.

Page last reviewed: 2023-08-18 Next review due: 2024-08-18

[Previous Maximum System Surplus](#) [Next Global Liquidation Limit](#) Last updated 6 months ago On this page * [Description](#) * [Purpose](#) * [Trade-offs](#) * [Enhanced Dai Savings Rate](#) * [Changes](#) * [Considerations](#)

Was this helpful? [Edit on GitHub](#)