# Flapper - Detailed Documentation

The Maker Protocol's Surplus Auction House * Contract Name: * flap.sol * Type/Category: * DSS —> System Stabilizer Module * Associated MCD System Diagram * Contract Source * Etherscan *

1. Introduction (Summary)

Summary: Flapper is a Surplus Auction. These auctions are used to auction off a fixed amount of the surplus Dai in the system for MKR. This surplus Dai will come from theStability Fees that are accumulated from Vaults. In this auction type, bidders compete with increasing amounts of MKR. Once the auction has ended, the Dai auctioned off is sent to the winning bidder. The system then burns the MKR received from the winning bid.

?

1. Contract Details

Flapper (Glossary)

- Flap
- 
    - surplus auction (selling stablecoins for MKR) [contract]
- wards [usr: address]
- -rely
- /deny
- /auth
- Auth Mechanisms [uint]
- Bid
- 
    - State of a specific Auction[Bid]
- 
    - bid
- 
        - quantity being offered for thelot
- 
    - (MKR) [uint]
- 
    - lot
- 
        - lot amount (DAI) [uint]
- 
    - guy
- 
        - high bidder [address]
- 
    - tic
- 
        - Bid expiry [uint48]
- 
    - end
- 
        - when the auction will finish [uint48]
- *
- bids (id: uint)
- 
    - storage of allBid
- s byid
- [mapping]
- vat
- 
    - storage of the Vat's address [address]
- ttl
- 
    - bid lifetime / max bid duration (default: 3 hours) [uint48]

- lot
-
    - lot amount (DAI) [uint]
- beg
-
    - minimum bid increase (default: 5%) [uint]
- tau
-
    - maximum auction duration (default: 2 days) [uint48]
- kick
-
    - start an auction / put up a new DAIlot
- for auction [function]
- tend
-
    - make a bid, thus increasing the bid size / submit an MKR bid (increasingbid
- ) [function]
- deal
-
    - claim a winning bid / settling a completed auction [function]
- gem
-
    - MKR Token [address]
- kicks
-
    - total auction count [uint]
- live
-
    - cage flag [uint]
- file
-
    - used by governance to setbeg
- ,ttl
- , andtau
- [function]
- yank
-
    - is used during Global Settlement to movetend
- phase auctions to theEnd
- by retrieving the collateral and repaying DAI to the highest bidder. [function]
- tick()
- -
- resets theend
- value if there has been 0 bids and the originalend
- has passed.
-

Parameters Set By Governance

The Maker Governance voters determine the surplus limit. The surplus auction is triggered when the system has an amount of Dai above that set limit.

- Parameters Set through
- file
- :
-
    - beg
-
    - ttl
-
    - tau
- *
-

Note: MKR governance also determines theVow.bump which sets theBid.lot for each Flap auction and theVow.hump which determines the surplus buffer.

Authorizations

auth - check whether an address can call this method [modifier function]

- rely
-
    - allow an address to call auth'ed methods [function]
- deny
-
    - disallow an address from calling auth'ed methods [function]
-

1. Key Mechanisms & Concepts

The mechanism begins with the MKR holders (Maker Governance Voters) of the system. MKR holders will specify the amount of surplus allowed in the system through the voting system. Once they come to an agreement on what it should be set to, surplus auctions are triggered when the system has a surplus of DAI above the amount decided during the vote. System surplus is determined in theVow when theVow hasno system debt and has accumulated enough DAI toexceed the Surplus auction size (bump ) plus the buffer (hump )

In order to determine whether the system has a net surplus, both the income and debt in the system must be reconciled. In short, any user can do this by sending theheal transaction to the system contract named the "Vow". Provided there is a net surplus in the system, the surplus auction will begin when any user sends theflap transaction to theVow contract .

Once the auction has begun, a fixed amount (lot ) of DAI is put up for sale. Bidders then complete for a fixedlot amount of DAI with increasingbid amounts of MKR. In other words, this means that bidders will keep placing MKR bid amounts in increments greater than the minimum bid increase amount that has been set (this is thebeg in action).

The surplus auction officially ends when the bid duration ends (ttl ) without another bid getting placedOR when auction duration (tau ) has been reached. At auction end, the MKR received for the surplus DAI is then sent to be burnt thereby contracting the overall MKR supply.

?

1. Gotchas (Potential source of user error)

Keepers

In the context of running a keeper (more infohere ) in order to perform bids within an auction, a primary failure mode could occur when a keeper specifies an unprofitable price for MKR.

- This failure mode is due to the fact that there is nothing the system can do to stop a user from paying significantly more than the fair market value for the token in an auction (this goes for all auction types,flip
- ,flop
- , andflap
- ).
- Keepers that are performing badly in aflap
- auction run the risk of overpaying MKR for the DAI as there is no upper limit to thebid
- size other than their MKR balance.
-

Bid Increments During an Auction

Duringtend ,bid amounts will increase by abeg percentage with each newtend . The bidder must know the auction'sid , specify the right amount oflot for the auction, bid at leastbeg % more than the last bid and must have a sufficient MKR balance.

One risk is "front-running" or malicious miners. In this scenario, an honest keeper's bid of [Past-bid +beg %] would get committed after the dishonest keeper's bid for the same, thereby preventing the honest keeper's bid from being accepted and forcing them to rebid with a higher price ((Past-bid + beg) + beg)). The dishonest keeper would need to pay higher gas fees to try to get a miner to put their transaction in first or collude with a miner to ensure their transaction is first. This could become especially important as the bid reaches the current market rate for MKR<>DAI.

Quick Example :

Thebeg could be set to 3%, meaning if the current bidder has placed a bid of 1 MKR, then the next bid must be at least 1.03 MKR. Overall, the purpose of the bid increment system is to incentivize early bidding and make the auction process move quickly.

Placing Bids Incorrectly

Bidders send MKR tokens from their addresses to the system/specific auction. If one bid is beat by another, the losing bid is

refunded back to that bidder's address. It's important to note, however, that once a bid is submitted, there is no way to cancel it. The only possible way to have that bid returned is if it is outbid (or if the system goes into Global Settlement).

Illustration of the bidding flow:

1. Vowkick
2. 's a new Flap Auction.
3. Bidder 1 sends a bid (MKR) that increases thebid
4. above the initial 0 value set during thekick
5. . Bidder 1's MKR balance is decreased and the Flap's balance is increased by the bid size.bid.guy
6. is reset from the Vow address to Bidder 1's andbid.tic
7. is reset tonow + ttl
8. .
9. Next, Bidder 2 makes a bid that increases Bidder 1's bid by at leastbeg
10. . Bidder 2's MKR balance is decreased and Bidder 1's balance is increased by Bidder 1'sbid
11. . The difference between Bidder 2's and Bidder 1'sbid
12. is sent from Bidder 2 to the Flap.
13. Bidder 1 then makes a bid that increases Bidder 2'sbid
14. by at leastbeg
15. . Bidder 1's MKR balance is decreased and Bidder 2's MKR balance is increased by Bidder 2'sbid
16. . The amount Bidder 1 increased the bid is then sent from Bidder 1 to the Flap.
17. Bidder 2, as well as all the other bidders participating within the auction, decide it is no longer worth it to continue to bid higherbid
18. s, so they stop making bids. Once theBid.tic
19. expires, Bidder 1 callsdeal
20. and the surplus DAI tokens are sent to the winning bidder's address (Bidder 1) in theVat
21. and the system then burns the MKR received from the winning bidder.gem.burn(address(this), bids[id].bid)
22. .
23.

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

1. SeeSystem Stabilizer Module Documentation

1. Other Failure Modes

2. Resulting from when MKR is burned

3.
    - There is the possibility where a situation arises where the MKR token makes the transaction revert (e.g. gets stopped or the Vow's permission to call burn() is revoked). In a case like this, deal can't succeed until someone fixes the issue with the MKR token. In the case of stoppage, this could include the deploying of a new MKR token. This new deployment could be completed by any individual using the MCD System but governance would need to add it to the system. Next, it would need to replace the old surplus and debt auctions with the new ones using the new MKR token. Lastly, it is crucial to enable the possibility to vote with the new version as well.
4. *
5. When there is massive surplus
6.
    - This would result in many Flap auctions occurring as the surplus overbump
7.
    - +hump
8.
    - is always auctioned off inbump
9.
    - increments. However, auctions run concurrently, so this would "flood the keeper market" and possibly result in too few bids being placed on any auction. This could happen through keepers not bidding on multiple auctions at once, which would result in network congestion because all keepers are trying to bid on all of the auctions. This could also lead to possible keeper collusion (if the capital pool is large enough, they may be more willing to work together to split it evenly at the system's expense).
10. *
11.

Previous System Stabilizer Module Next Flopper - Detailed Documentation Last updated3 years ago On this page *

Export as PDF