

MCD Glossaries

A list of words, terms, variables, functions and more relating to the Maker Protocol

1. [MCD General Glossary of Terms](#)

2. MCD Core Smart Contracts Glossary:

General

- guy
- ,usr
- : some address
- wad
- : some quantity of tokens, usually as a fixed point integer with 18 decimal places.
- ray
- : a fixed point integer, with 27 decimal places.
- rad
- : a fixed point integer, with 45 decimal places.
- file
- : administer some configuration value
-

Auth

- auth
- : check whether an address can call this method
- ward
- : an address that is allowed to call auth'ed methods
- rely
- : allow an address to call auth'ed methods
- deny
- : disallow an address from calling auth'ed methods
- Authority
- - checks whether an address can call this method
- Kiss
- - cancels out surplus and on-auction debt
-

Vat (Vault Engine)

- gem
- : collateral tokens.
- dai
- : stablecoin tokens.
- sin
- : unbacked stablecoin (system debt, not belonging to anyurn
-).
- ilk
- : a collateral type.
- - rate
- - : stablecoin debt multiplier (accumulated stability fees).
- - take
- - : collateral balance multiplier.
- - Ink
- - : total collateral balance.
- - Art
- - : total normalized stablecoin debt.

- *
- init
- : create a new collateral type.
- urn
- : a specific Vault.
- - ink
- - : collateral balance.
- - art
- - : normalized outstanding stablecoin debt.
- *
- slip
- : modify a user's collateral balance.
- flux
- : transfer collateral between users.
- move
- : transfer stablecoin between users.
- grab
- : liquidate a Vault.
- heal
- : create / destroy equal quantities of stablecoin and system debt (vice
-).
- fold
- : modify the debt multiplier, creating / destroying corresponding debt.
- toll
- : modify the collateral multiplier, creating / destroying corresponding collateral.
- suck
- : mint unbacked stablecoin (accounted for with vice
-).
- spot
- : collateral price with safety margin, i.e. the maximum stablecoin allowed per unit of collateral.
- line
- : the debt ceiling for a specific collateral type.
- Line
- : the total debt ceiling for all collateral types.
- dust
- : the minimum possible debt of a Vault.
- frob
- : modify a Vault.
- - lock
- - : transfer collateral into a Vault.
- - free
- - : transfer collateral from a Vault.
- - draw
- - : increase Vault debt, creating Dai.
- - wipe
- - : decrease Vault debt, destroying Dai.
- - dink
- - : change in collateral.
- - dart
- - : change in debt.
- - calm

- - : true when the Vault remains under both collateral and total debt ceilings.
- - cool
- - : true when the stablecoin debt does not increase.
- - firm
- - : true when the collateral balance does not decrease.
- - safe
- - : true when the Vault's ratio of collateral to debt is above the collateral's liquidation ratio.
- *
- fork
- : to split a Vault - binary approval or splitting/merging Vaults.
- - dink
- - : amount of collateral to exchange.
- - dart
- - : amount of stablecoin debt to exchange.
- *
- wish
- : check whether an address is allowed to modify another address's gem or dai balance.
- - hope
- - : enablewish
- - for a pair of addresses.
- - nope
- - : disablewish
- - for a pair of addresses.
- *
-

Note: art and Art represent normalized debt, i.e. a value that when multiplied by the correct rate gives the up-to-date, current stablecoin debt.

Accounting

- debt
- : the sum of all dai
- (the total quantity of dai issued).
- vice
- : the sum of all sin
- (the total quantity of system debt).
- ilk.Art
- : the sum of all art
- in the urn
- s for that ilk
- .
- debt
- : is vice
- plus the sum of $\text{ilk.Art} * \text{ilk.rate}$
- across all ilk
- 's.
-

Collateral

- gem

- : can always be transferred to any address by it's owner.
-

Dai

- dai
- : can only move with the consent of it's owner / can always be transferred to any address by it's owner.
-

Other

- LogNote
- : a general purpose log that can be added to any function from a contract.
-

Jug (Stability Fees)

Structs

Ilk : contains twouint256 values—duty , the collateral-specific risk premium, andrho , the timestamp of the last fee update

VatLike : mock contract to make Vat interfaces callable from code without an explicit dependency on the Vat contract itself

Storage

wards :mapping(address => uint) that indicates which addresses may call administrative functions

ilks :mapping (bytes32 => Ilk) that stores anIlk struct for each collateral type

vat : aVatLike that points the the system's [Vat](#) contract

vow : theaddress of the Vow contract

base : auint256 that specifies a fee applying to all collateral types

Administrative

These methods requirewards[msg.sender] == 1 (i.e. only authorized users may call them).

rely /deny : add or remove authorized users (via modifications to thewards mapping)

init(bytes32) : start stability fee collection for a particular collateral type

file(bytes32, bytes32, uint) : setduty for a particular collateral type

file(bytes32, data) : set thebase value

file(bytes32, address) : set thevow value

Fee Collection Methods

drip(bytes32) : collect stability fees for a given collateral type

Cat (Liquidations)

- mat
- : the liquidation ratio
- chop
- : the liquidation penalty
- lump
- : the liquidation quantity, i.e. the fixed debt quantity to be covered by any one liquidation event
- bite
- : initiate liquidation of a Vault
- flip
- : liquidate collateral from a Vault to cover a fixed quantity of debt
-

Vow (Settlement)

- sin
- : the system debt queue.

- Sin
- : the total amount of debt in the queue.
- Ash
- : the total amount of on-auction debt.
- wait
- : length of the debt queue
- sump
- : debt auction bid size, i.e. the fixed debt quantity to be covered by any one debt auction
- dump
- : debt auction lot size, i.e. the starting amount of MKR offered to cover the lot
- /sump
- bump
- : surplus auction lot size, i.e. the fixed surplus quantity to be sold by any one surplus auction
- hump
- : surplus buffer, must be exceeded before surplus auctions are possible
-

Other terms included in Vow documentation:

- move
- : transfers stablecoin between users.
- kick
- : starts an auction.
- woe
- : indicates specifically bad debt, or be used as a variable name for any amount of debt.
-

Flipper (Collateral Auctions)

- wards [usr: address]
- ,rely
- /deny
- /auth
- : Auth mechanisms
- Bid
- : State of a specific Auction {bid
- ,lot
- ,guy
- ,tic
- ,end
- ,usr
- ,gal
- ,tab
- }
-
- - bid
- - : Bid amount (DAI)/ DAI paid
- - lot
- - : quantity up for auction / collateral gems for sale
- - guy
- - : high bidder (address)
- - tic
- - : Bid expiry
- - end
- - : when the auction will finish / max auction duration
- - usr
- - : address of the Vault being auctioned. Receives gems during the debt
-

- phase
-
- gal
-
- : recipient of auction income / receives dai income (this is the Vow contract)
-
- tab
-
- : total dai wanted from the auction / total dai to be raised (in flip auction)
- *
- bids[id: uint]
- : storage of all bids
- vat
- : storage of the Vat's address
- ilk
- : id of the Ilk for which the Flipper is responsible
- beg
- : minimum bid increase (default: 5%)
- ttl
- : bid duration (default: 3 hours)
- tau
- : auction length (default: 2 days)
- kicks
- : Total auction count, used to track auctionid
- s
- kick
- : function used byCat
- to start an auction / Put collateral up for auction
- tick
- : restart an auction if there have been 0 bids and theend
- has passed
- tend
- : first phase of an auction. Increasing Daibid
- s for a setlot
- of Gems
- dent
- : second phase of an auction. Set Daibid
- for a decreasinglot
- of Gems
- file
- : function used by governance to setbeg
- ,ttl
- , andtau
- deal
- : claim a winning bid / settles a completed auction
- yank
- : used during Global Settlement to movetend
- phase auctions to theEnd
- by retrieving the collateral and repaying dai to the highest bidder.
-

Flapper (Surplus Auctions)

- 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 DAI lot
- 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.
-

Flopper (Debt Auctions)

- flop
- : debt auction (covering debt by inflating MKR and selling for stablecoins)
- lot
- : quantity up for auction / gems for sale (MKR)
- guy
- : high bidder (address)
- gal
- : recipient of auction income / receives dai income (this is the Vow contract)

- ttl
- : bid lifetime (Max bid duration / single bid lifetime)
- beg
- : minimum bid decrease
- pad
- : Increase forlot
- size duringtick
- (default to 50%)
- tau
- : maximum auction duration
- end
- : when the auction will finish / max auction duration
- kick
- : start an auction / Put up a new MKRbid
- for auction
- dent
- : make a bid, decreasing the lot size (Submit a fixed DAIdbid
- with decreasinglot
- size)
- deal
- : claim a winning bid / settles a completed auction
- vat
- : the Vat's address
- gem
- : MKR Token (address)
- kicks
- : Total auction count, used to track auctionid
- s
- live
- : Cage flag
- wards [usr: address]
- ,rely
- /deny
- /auth
- : Auth mechanisms
- Bid
- : State of a specific Auction {bid
- ,lot
- ,guy
- ,tic
- ,end
- }
- bid
- : Bid amount inDAI / DAI paid
- tic
- : Bid expiry
- tick
- : restarts an auction
-

End (Global Settlement / Shutdown)

cage : Locks the system and initiates shutdown. This is done by freezing the user-facing actions, canceling flap and flop auctions, locking the rest of the system's contracts, disabling certain governance actions that could interfere with the settlement process, and starting the cool-down period.

cage(ilk) : Tags the ilk prices / Sets the final price for an ilk (tag).

skim : Settles a Vault at the tagged price / Cancels owed Dai from the Vault

free : Remove (remaining) collateral from a settled Vault. This occurs only after there is no debt in the Vault.

thaw : Fixes the Dai supply after all Skims / Fixes the total outstanding supply of stablecoin.

flow : Calculates the fixed price for an ilk, possibly adjusting the cage price with surplus/deficit.

pack : Locks Dai ahead of Cash / Puts some stablecoin into abag in preparation for cash.

cash : Exchange packed Dai for collateral / Exchange some Dai from bag for a given gem, share proportional to bag size.

file : The Governance configuration—sets various parameter values.

skip : optionally cancel live auctions.

Other

wards(usr: address) : Auth Mechanism

vat : Vat contract

cat : Cat contract

vow : Vow contract

spot : Spotter contract

live : Cage flag

- "Live" contracts have live
- = 1, indicating the system is running normally. Thus, when cage()
- is invoked, it sets the flag to 0. This includes the End
- contract, which means that cage()
- can only be invoked once and the subsequent functions cannot be invoked until we are "dead" and in the End process
-

ilk : A collateral type

when : Time of cage / the time of settlement.

wait : Processing cooldown duration / the length of processing cooldown.

debt : Outstanding Dai after processing / outstanding stablecoin supply, after system surplus/deficit has been absorbed.

tag : Cage price / price per collateral type at time of settlement.

gap : Collateral shortfall / shortfall per collateral considering undercollateralised Vaults.

Art : Total debt per ilk/outstanding stablecoin debt.

fix : Final cash price / the cash price for an ilk (amount per stablecoin).

bag(usr: address) : Dai packed for cash / nontransferable stablecoins ready to exchange for collateral.

out : Cash out / the amount of already exchanged stablecoin for a given address.

skip : Optionally cancel live auctions.

wad : Some quantity of tokens, usually as a fixed point integer with 10^{18} decimal places.

urn : A specific Vault.

tend : To make a bid, increasing the bid size.

bid : The quantity being offered for the lot .

lot : The quantity up for auction.

dent : To make a bid, decreasing the lot size.

Join (Token Adapters)

- vat
- : storage of the Vat
- 's address.
- ilk
- : id of the ilk for which aGemJoin
- is created for.
- gem
- : the address of the ilk
- for transferring.
- dai
- : the address of the dai

- token.
- one
- : a 10^{27} uint used for math in DaiJoin
- .
- live
- : an access flag for the join
- adapter.
- dec
- : decimals for the Gem.
-

Cat (Liquidations)

- mul(uint, uint)
- /rmul(uint, uint)
- : will revert on overflow or underflow
- bite(bytes32, address)
- : will revert if not
- orat
- are larger than or equal to 2^{255} .
- bite
- : will not leave a Vault with debt and no collateral.
- wards
- : are allowed to call protected functions (Administration and cage())
-)
- ilks
- : stores Ilk
- structs
- - Ilk
- - is the struct with the address of the collateral auction contract (flip
- -), the penalty for that collateral to be liquidated (chop
- -) and the maximum size of collateral that can be auctioned at once (lump
- -).
- *
- live
- : must be 1
- for the Cat
- to bite
- (see cage
- in mechanisms)
- vat
- : address that conforms to a VatLike
- interface (see vat
- documentation [TODO - Link] for more info). It is set during the constructor and cannot be changed
- .
- vow
- : address that conforms to a VowLike
- interface (see vow
- documentation [TODO - Link] for more info).
-

Events

- Bite
- : emitted when abite(bytes32, address)
- is successfully executed. Contains:
 - ilk
- - : Collateral
- - urn
- - : Vault address

- - ink
- - : seelot
- - inbite
- - art
- - : seeart
- - inbite
- - tab
- - : seetab
- - inbite
- - flip
- - : address of the auction contract
- - id
- - : ID of the auction in theFlipper
- *
-

Spot (Oracles and Contracts Liaison)

- ilk
- : a given collateral type
- ilk.pip
- : the contract which holds the current price of a givenilk
- ilk.mat
- : the liquidation ratio for a givenilk
- vat
- : the core of the mcd system
- par
- : the relationship between DAI and 1 unit of value in the price. (Similar to TRFM)
-

Collateral

- Only authorized users can update any variables in contract
-

Pot (Savings Dai)

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
- .
-

Auth

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

Storage

- pie
- : stores the address'Pot
- balance.
- Pie
- : stores the total balance in thePot
- .
- dsr
- : the dai savings rate
- . It starts as 1
- ($ONE = 10^{27}$
-), but can be updated by governance.
- chi
- : the rate accumulator. This is the always increasing value which decides how much dai
- : given when drip()
- is called.
- vat
- : an address that conforms to a VatLike
- interface. It is set during the constructor and cannot be changed
- .
- vow
- : an address that conforms to a VowLike
- interface. Not set in constructor. Must be set by governance.
- rho
- : the last time that drip is called.
-

Dai (Stablecoin)

name : Dai Stablecoin

symbol : DAI

version : 1

decimals : 18

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

totalSupply : Total DAI Supply

balanceOf(usr: address) : User balance

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

nonces(usr: address) : Permit nonce

[Previous ESM - Detailed Documentation](#) [Next Smart Contract Annotations](#) Last updated 2 years ago On this page * [1. MCD General Glossary of Terms](#) * [2. MCD Core Smart Contracts Glossary:](#) * [General](#) * [Auth](#) * [Vat \(Vault Engine\)](#) * [Jug \(Stability Fees\)](#) * [Cat \(Liquidations\)](#) * [Vow \(Settlement\)](#) * [Flipper \(Collateral Auctions\)](#) * [Flapper \(Surplus Auctions\)](#) * [Flopper \(Debt Auctions\)](#) * [End \(Global Settlement / Shutdown\)](#) * [Join \(Token Adapters\)](#) * [Cat \(Liquidations\)](#) * [Spot \(Oracles and Contracts Liaison\)](#) * [Pot \(Savings Dai\)](#) * [Dai \(Stablecoin\)](#)

[Export as PDF](#)