# SCD <> MCD Migration

The Maker Protocol Upgrade Contract * Contract Name: * scd-mcd-migration.sol * Type/Category: * Migration *Associated MCD System Diagram * Contract Source * Etherscan *

1. Introduction (Summary)

The Migration contract's purpose is to allow moving SAI and CDPs from the SCD system to the MCD system, thus becoming DAI and Vaults. It also allows users to move SAI/DAI in both directions should they want to exit MCD and go back to SCD.

?

?

1. Contract Details

Key Functionalities

swapSaiToDai - Takes Sai (ERC-20 DAI from Single Collateral System) returns DAI (ERC-20 DAI from MultiCollateral System).

swapDaiToSai - Takes DAI (ERC-20 DAI from MultiCollateral System) returns Sai (ERC-20 DAI from Single Collateral System)

migrate - Moves a Vault from SCD to MCD by closing the SCD one and opening a corresponding MCD one.

Storage

tub : SCD Tub contract address

vat : MCD Vat Contract address

cdpManager : MCD CDP manager

saiJoin : SAI collateral adapter for MCD

wethJoin : WETH collateral adapter for MCD

daiJoin : DAI join adapter for MCD

1. Key Mechanisms & Concepts

Overall this contract has two primary purposes:

1. Two-way exchange for SAI and DAI
2. One-way transfer of Vaults
3.

1.constructor (setup)

2.swapSaiToDai

Users of the current DAI system will want to move to the new MCD system. This function allows DAI holders to seamlessly convert their DAI. They will need toapprove the migration contract on the SAI ERC-20 contract so that it can perform thetransferFrom . The migration contract holds a Vault in MCD that takes SAI as collateral and allows it to exit MCD-DAI, which it does and returns to themsg.sender .

3.swapDaiToSai

In case a user wants to go back to SAI, this function allows them to turn in their DAI in exchange for SAI. This requires the userapprove s the migration contract on the DAI ERC-20 contract so that it cantransferFrom thenjoin the DAI back into MCD. This pays back the "debt" in its SAI-MCD Vault and allows it to retrieve the SAI "collateral" and return it to themsg.sender .

4.migrate

This function is meant to be used in combination with theMigrationProxyActions as it requires the migration contract owns the SCD-Vault (cup ) already and that the migration contract has enough MKR to pay the stability fees. TheMigrationProxyActions migrate functiontransferFrom s themsg.sender to the migration contract so that the migration contract has enough MKR to pay the stability fee and close thecup .

The migration contract first draws its own SAI out of its MCD contract and uses that to pay back the debt for thecup (along with the MKR it has from the proxy action to pay the fee). Then it withdraws the PETH as WETH.

Next the migration contract opens a Vault using the MCD CDP manager andjoin s its WETH into its new Vault and withdraws enough DAI from the new Vault (and pays back its Vault) to compensate for the SAI it drew earlier in this step.

Lastly, the migration contract gives the MCD-Vault to themsg.sender .

1. Gotchas (Potential source of user error)

2. Any special/unique information about the specific contract

3. Anything that it may rely on, especially if it is not obvious
4. Sources of user error if not explicitly defined
5.

swapSaiToDai

Thewad amount has to be below the debt ceiling for both the overall MCD system and the SAI collateral type, otherwise thefrob will fail. This means that these governance parameters can impact the speed of the transition from SAI to DAI.

swapDaiToSai

Thewad amount has to be below the amount of SAI collateral in the migration contract's Vault. If a user with DAI wants to move to SAI but no SAI users have already moved to DAI, then this will fail.

migrate

Because the migration contract will have to first draw SAI from its MCD collateral, the system will have to be seeded with SAI in the migration contract's Vault in an amount that exceeds the SAI debt for thecup being migrated.

If a user holds both acup and SAI, they should decide whether it makes sense to:

1. Pay back thecup
2. in SCD, thenmigrate
3. theircup
4. to MCD (essentially just transfer the collateral to a new MCD Vault).
5. migrate
6. theircup
7. with the debt in place, then useswapSaiToDai
8. to get DAI which they can then use as an ERC-20 or payback their MCD debt.
9.

One additional consideration, to close or migrate acup , a user will have to purchase MKR in order to pay the stability fee and be able toexit the SCD system. However, once in MCD, new fees will be accrued (and have to be paid) in DAI. If a user's converted SAI does not cover their MCD debt + stability fee, they may have to purchase DAI on the open market.

Before SCD shutdown: Users who took out a Vault in SCD and then used the DAI to purchase something will either have to buy SAI on the open market to pay back their SCD debt or they will have to migrate their collateral to MCD.

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

2. Potential for error:

3. Governance parameters around SAI collateral
4. 
    - Collateralization ratio has to be set to a very low number
5. 
    - Bothilks["sai"].duty
6. 
    - andJug.base
7. 
    - have to be set to0
8. 
    - during the migration period
9. *
10. Auth errors on Sai Join
11. Excess Sai in MCD (i.e. morecup

12. s are lost/not migrated than lost/not migrated Sai): results in an auction and possibly MKR auction to cover bad debt.
13.

Migration

- Sai debt ceiling to 0
- MCD.ilks[sai] debt ceiling to SCD.totalDai
- DSR value competitive with Compound to encourage migration
-

Previous Flipper - Detailed Documentation Next Upgrading to Multi-Collateral Dai Guide Last updated4 years ago On this page * 1. Introduction (Summary) * 2. Contract Details * 3. Key Mechanisms & Concepts * 4. Gotchas (Potential source of user error) * 5. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

Export as PDF