

Upgrading to Multi-Collateral Dai Guide

Summary

Level: Intermediate

Estimated Time: 30 minutes

Audience: Technical and commercial teams with partners and Dai holders

Introduction

The upcoming version of the Maker system, Multi-Collateral Dai, brings a lot of new and exciting features, such as support for new Vault collateral types and Dai Savings Rate. In order to support the new functionality, the whole Maker core of smart contracts has been rewritten. The new smart contracts addresses and ABIs can be found here: <https://changelog.makerdao.com/releases/mainnet/1.0.0/>

Therefore, users and partners interacting with Single-Collateral Dai (SCD) must migrate their existing Single Collateral Dai tokens (Sai) to Multi Collateral Dai tokens (Dai) and CDPs to the new system. Additionally, companies or projects integrated with Sai and CDPs must update their codebases to point to the new smart contracts, and refactor their code to support the updated functions.

This guide will focus on the Dai and CDP migration with a high level overview of the upgrade process for different actors in the Maker ecosystem.

The steps necessary to migrate from Single-Collateral Dai (SCD) to Multi-Collateral Dai (MCD) differ depending on your platform and use case for Dai, so the guide is split into sections for different user and partner types.

Important note on naming conventions

In this guide we refer to the Single Collateral Dai system as SCD, and the Multi-Collateral Dai system as MCD. We refer to the Single Collateral Dai token (the old, currently existing Dai) as Sai, and the new Multi-Collateral Dai token as Dai.

Learning Objective

- Knowledge on how migration to MCD will work
- Best practices for migration for different users and partners
- Where to find guides on specific migration scenarios
-

Pre-requisites

- Basic knowledge of the MakerDAO: Dai and/or Vault system [See the MCD 101 guide, especially sections 1 and 2.](#)
-

Sections

- [User and Partner Migration Scenarios](#)
-
- [As a Sai Holder](#)
-
- [As a SCD CDP Owner](#)
-
- [As a Centralized Exchange or Custodial Wallet](#)
-
- [As a Decentralized Exchange](#)
-
- [As a Non-Custodial Wallet](#)
-
- [As a Keeper](#)
-
- [As a Market Maker](#)
-
- [As a CDP Integrator](#)
-
- [As a Lending Protocol](#)
-
- [As a Dapp](#)
-

- [As another partner type not mentioned above](#)
- *
- [Migration App](#)
- [Migration Contract](#)
-
- [Functionality](#)
-
- [Upgrading Sai to Dai](#)
-
- [Swapping Dai for Sai](#)
-
- [Migrating CDPs](#)
- *
-

User and Partner Migration Scenarios

The following section will outline a recommended migration process for different actors in the Maker ecosystem.

As a Sai Holder

You control your private key

If you hold your Sai in a wallet where you control your private keys, then head to migrate.makerdao.com (available at MCD launch) and follow the instructions to upgrade your Sai to Dai and optionally activate the Dai Savings Rate smart contract, which allows you to earn savings.

The following figure outlines the migration flow:

You don't control your private key

If your Sai is deposited in an exchange or centralized wallet or locked in a dApp smart contract, you can follow the instructions these platforms are providing or withdraw the Sai and complete the upgrade yourself at migrate.makerdao.com

With MCD you can deposit your Dai into the Dai Savings Rate smart contract which will earn you accrued annual savings. Find more info at makerdao.com at launch.

As a SCD CDP Owner

As a SCD CDP owner you can move your CDP to the MCD CDP core through the Migration App at migrate.makerdao.com at launch. The following diagram shows the flow for CDP migration.

You can also choose to manually close your CDP by paying back your debt and redeeming your Ether, and use your redeemed collateral to open a new MCD CDP.

If you have a large SCD CDP, the migration contract might not have enough Sai liquidity to carry out the migration. In that case, feel free to contact integrate@makerdao.com for assistance. You can read more about migration in the [Migration Contract](#) section later in this guide.

Notes on Instadapp

If you have created your CDP through the Instadapp service, you need to withdraw ownership of the CDP from the service back to you. To do this, you need to navigate to the [exit page](#) and click "Withdraw" on your CDP in the tab "Debt Positions". This will give you custody of the CDP, which will make it visible at migrate.makerdao.com where you will be able to carry out CDP migration.

Notes on MyEtherWallet

If you have created your CDP on MyEtherWallet then you can migrate your CDP using the Migration App at migrate.makerdao.com. (However, if the private key used with MyEtherWallet is stored in a local file or another unsupported format, you must first import your key to a wallet with Web3 support.)

Once upgraded, you can start using Dai Savings Rate by locking your Dai into the Dai Savings Rate smart contract and receive accrued savings. Find more info on makerdao.com at launch.

As a Centralized Exchange or Custodial Wallet

We recommend you take the following steps for upgrading to MCD:

- On November 18: Rename Single-Collateral Dai to "Sai". This is being coordinated with all Maker partners and serves to avoid users depositing the wrong token into your system.

- On December 2: Perform upgrade of user balances.
- Inform your users as soon as possible about the dates. For users wanting to delay their upgrade, this allows them to opt-out by withdrawing Sai from your exchange before the date.
- Proposed process for the December 2 upgrade:
 -
 - Freeze Sai deposits/withdrawals
 -
 - Use the Migration App/contract to upgrade all user holdings of Sai to Dai. (See more details in the [Migration App](#)
 -
 - [/contract](#)
 -
 - sections below.)
 -
 - Point codebase to new Dai token contract address. The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](#)
 -
 - - use the [updated logos found here](#)
 -
 - for the new Dai token.
 -
 - Rename listing/token to "Dai"
 -
 - Unfreeze Dai deposits/withdrawals.
 - *
 - Inform users about [Dai Savings Rate](#)
 - , which allows Dai holders to earn savings.
 -
 - Optional: Choose one of the following:
 -
 - - Integrate Dai Savings Rate and distribute revenue to your users.
 -
 - - Integrate Dai Savings Rate in your exchange and keep accrued savings in your own balance sheet.

-
- *
 -

This approach will result in the following user journey for the exchange/wallet user:

As a Decentralized Exchange

We recommend you take the following steps for upgrading to MCD:

- On November 18: Rename Single-Collateral Dai to "Sai" and ticker "SAI". This is being coordinated with all Maker partners and serves to avoid users attempting to deposit the wrong token into your system.
- Select a date between November 18-25 to list Multi-Collateral Dai. The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](#)
- - use the [updated logos found here](#)
- for the new Dai token. Logo for Sai should remain the yellow diamond.
- On the date of your own Dai listing: Add support for the new Dai token on your platform. The new Dai token should be named Dai and have the ticker "DAI". Deactivate Sai trading in your frontend UI, but allow users to cancel orders and withdraw balances.
- Inform users that they can redeem Sai for Dai at [migrate.makerdao.com](#)
- - Optional: Provide a UI in your own interface for token migration through the migration contract.
- *
- Inform users about Dai Savings Rate, which allows Dai holders to earn savings.
- - Optional: Build a UI that facilitates the usage of the Dai Savings Rate service for your users in your exchange, where users will keep the accrued savings themselves.
-
- Optional: Link users to [oasis.app](#)
-
- to activate Dai Savings Rate.
- *

As a Non-Custodial Wallet

If you are a creator of a wallet that allows users to be in control of their private keys we recommend you do the following:

- On November 18: Rename Single-Collateral Dai to “Sai”
- Select a future date between November 18-25 to execute the upgrade to support Multi-Collateral Dai, which should be listed as “Dai”. The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](https://etherscan.io/address/0x6b175474e89094c44da98b954eedeac495271d0f)
- - use the [updated logos found here](#)
- for the new Dai token. Logo for Sai should remain the yellow diamond.
- Inform your users as soon as possible about the timeline for your own upgrade to MCD.
- Support balances of both Sai and Dai for a period until Sai demand diminishes.
- Inform your users that they will be able to swap Sai for Dai at migrate.makerdao.com
-
- - Optional: Provide a UI in your own interface for token migration through the migration contract.
- *
- Inform users about Dai Savings Rate, which allows Dai holders to earn savings.
- - Optional: Create a UI where users can activate Dai Savings Rate.
-
- - Optional: Link users to oasis.app
 - to activate Dai Savings Rate.
- *
- Optional: Implement paying the gas cost of Dai transactions on behalf of your users.
-

As a Keeper

- Get acquainted with the updates to Keepers and Auctions in MCD with [this guide](#)
-
- Upgrading
- We expect to release a Python library for working with Auctions before MCD launch. This will be the recommended way to bid in Auctions.
-

Alternatively, if you’re willing to do some additional work and work with a lower level interface, you can interact with Auction contracts directly ([flip](#), [flap](#), [flop](#)). Note that future collateral types may come with custom auction formats. More documentation will be available before launch.

As a Market Maker

- We encourage you to market make on Multi-Collateral Dai as soon as your exchange partners add support for it.
- If your exchange partners keep their Sai listing concurrently with their Dai listing, we encourage you to market make on both tokens for the remaining lifetime of Sai.
- If your exchange partners will use a different ticker for Dai than Sai, you should update your tools accordingly.
-

As a CDP Integrator

Custodial CDP service

- On November 18: Rename Single-Collateral Dai to “Sai”
- Select a future date between November 18-25 to execute the upgrade to MCD.
- Inform your users as soon as possible about the date.
- On the chosen date:
- - Freeze access to CDP service for your users
- - Launch upgrade to your service that supports the new CDP core [The smart contract addresses and ABIs can be found here.](#)
- - - If you are using Dai.js for your CDP integration, see [Using Dai.js](#)
- - - ” below for how to upgrade your implementation to MCD.

- - - If you have integrated directly with the CDP smart contracts, see [Direct integration with smart contracts](#)
 - - - " below for how to upgrade your implementation to MCD.
-

- - Migrate all CDPs to MCD. See "Migration App" section below.
- - List the Multi-Collateral Dai token as "Dai"
- - Unfreeze access to CDP service
- *
- Optional: Implement support for added collateral types in MCD
- If it is relevant to your service, inform users about Dai Savings Rate
- - Optional: Implement UI for locking Dai in the Dai Savings Rate smart contract.
- *
-

Non-Custodial CDP service

- On November 18: Rename Single-Collateral Dai to "Sai"
 - Select a future date between November 18-25 to execute the upgrade to MCD.
 - Inform your users as soon as possible about the timeline for your own upgrade to MCD.
 - Inform your users about MCD and the migration process of CDPs.
 - On the selected launch date:
 - Launch upgrade to your service that supports the new CDP core.
 - - - If you are using Dai.js for your CDP integration, see [Using Dai.js](#)
 - - - " below for how to upgrade your implementation to MCD.
 - - - If you have integrated directly with the CDP smart contracts, see [Direct integration with smart contracts](#)
 - - - " below for how to upgrade your implementation to MCD.
-

- - List the Multi-Collateral Dai token as "Dai"
- *
- Choose one of the following:
 - Option A: Point your users to migrate.makerdao.com
 - at MCD launch date for CDP migration on their CDP dashboard. See also the Migration App section below.
 - Option B: Create your own UI for migration, by creating a frontend to interact with the migration contract (see section below on Migration Contract).
- *
-

Upgrading your CDP integration implementation

Using Dai.js

- If you have integrated CDPs using the [Dai.js library](#)
- , ensure you have updated the library to the latest version.
- Update your codebase to support the functionality of the [MCD plugin](#)
- . At launch this plugin will be bundled into the Dai.js library as default.
- Optional: Help your users migrate their CDP to MCD
-

- Option A: Point users to migrate.makerdao.com
-
- if your app is Web3 compatible.
-
- Option B: Implement your own migration UI in your app, connecting to the migration contract described in a section below.
-
- Option C: If your app is not compatible with migrate.makerdao.com, you can guide your users in how to export their CDP from your app to a compatible wallet.
- *
- Optional: Implement support for new MCD functionality
-
- Add support for new collateral types.
-
- Add support for Dai Savings Rate.
- *
-

Direct integration with smart contracts

- If you have integrated directly with the smart contracts, you must add support for the new Maker core smart contracts. Since the smart contracts have been completely rewritten, most function calls have been changed.
- Get acquainted with the [new implementation of MCD](#)
-
- [You can find an introduction to the system here](#)
- *
- Implement support for the MCD smart contracts
-
- [Checkout this guide on how to interact with the CDP manager.](#)
- *
- Point codebase to the new [MCD smart contracts](#)
-

As a Lending Protocol

Custodial Service

- On November 18: Rename Single-Collateral Dai to “Sai”
- Select a future date between November 18-25 to execute the upgrade to MCD.
- Inform your users as soon as possible about the date.
- On the chosen date:
-
- Stop lending (deposits) and borrowing (withdrawals) of Sai
-
- List the Multi-Collateral Dai token as "Dai". The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](https://etherscan.io/address/0x6b175474e89094c44da98b954eedeac495271d0f)
-
-
- use the [updated logos found here](#)
-
- for the new Dai token.
-
- Open for lending (deposits) and borrowing (withdrawals) of Dai
-
- For outstanding loans in Sai, choose one of the following:
-
-
- Accept payback of loans in Sai.
-
-
- Continuously migrate paybacks of old positions of Sai to Dai yourself.
-
-
- Inform your users that you can no longer pay back Sai, but that they should migrate their Sai to Dai through migrate.makerdao.com before paying back a loan.

-
- *
 -

Non-Custodial Service

- On November 18: Rename Single-Collateral Dai to “Sai”
- Select a future date between November 18-25 to execute the upgrade to MCD.
- Inform your users as soon as possible about the timeline for your own upgrade to MCD.
- Inform users about potential cutoff dates for shutdown of SCD.
- At launch:
 - - List the Multi-Collateral Dai token as "Dai". The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](#)
 - - - use the [updated logos found here](#)
 - - for the new Dai token.
 - - Launch support for Dai loans.
 - - Stop creation of loans in Sai
 - - Point users to [migrate.makerdao.com](#)
 - - for Sai migration
 - - Let existing loans in Sai run until they expire or are paid back
- *
- Optional:
 - Create a UI for users to migrate their balances from Sai to Dai.
- *
-

As a Dapp

- On November 18: Rename Single-Collateral Dai to “Sai”
- Select a future date between November 18-25 to execute the upgrade to MCD.
- Inform your users as soon as possible about the timeline for your own upgrade to MCD.
- On the chosen date:
 - - List the Multi-Collateral Dai token as "Dai". The new token is deployed at [0x6b175474e89094c44da98b954eedeac495271d0f](#)
 - - - use the [updated logos found here](#)
 - - for the new Dai token.
 - - Update code base to support the use of the new Dai token at launch.
 - - Optional: Implement paying gas cost of Dai transactions in Dai.
- *
- If you have a product using Sai:
 - Shutdown functionality of Sai at a cut-off date, communicated well in advance to your users.
- *
- Inform your users about potential confusion regarding Sai and Dai.
- Inform your users that they can migrate Sai to Dai at [migrate.makerdao.com](#)
- - Optional: Create a UI for carrying out the migration from Sai to Dai.
- *
-

As another partner type not mentioned above

Please reach out to integrate@makerdao.com and we are happy to discuss your migration scenario.

Migration App

Upon release of MCD, the Migration App at [migrate.makerdao.com](#) will allow you to carry out Dai and CDP migration through

an intuitive web UI in just a few clicks. By logging in with your favourite wallet solution, the app will scan your wallet for any recommended migrations and showcase them in the UI (seen in picture below). This migration scan feature is planned to be continually supported going forward, ensuring that users are always using an up-to-date version of the Maker platform.

Landing Page that will show you possible migrations for the connected wallet.

?

Wizard for migrating Sai to Dai.

?

Wizard for migrating an SCD CDP to MCD CDP.

The Migration App uses a proxy contract to carry out the CDP migration. Consequently, the app can also only be used for CDPs that have been created through a Maker proxy contract. This happens automatically if you have opened your CDP at cdp.makerdao.com.

If you have created CDPs using third party services that do not use Maker proxies to interact with the CDP core, the migration contract might not work. Instead, you can perform your own manual migration by simply closing down your SCD CDP and moving the ETH to an MCD CDP.

Migration Contract

The functionality of the Migration App outlined in the above section is handled by a Migration Contract that will be deployed at MCD launch in order to support a smooth transition from Single Collateral Dai to Multi Collateral Dai. The contract will make the redemption of Single Collateral Dai tokens (Sai) for Multi Collateral Dai tokens (Dai), and the migration of CDPs to the new CDP engine of MCD, an easy task. This section will outline how the migration contract works in order to help super users and partners prepare for MCD migration.

Functionality

The migration smart contracts are open source and can be found here <https://github.com/makerdao/scd-mcd-migration>

In the src folder, the smart contract source code can be found. The main contract is the ScdMcdMigration.sol which contains the migration functionality that you will interact with.

It contains three main functions:

- swapSaiToDai
 - a function that upgrades Sai to Dai
- swapDaiToSai
 - a function that allows you to exchange your Dai back to Sai
- migrate
 - a function that allows you to migrate your SCD CDP to MCD.

The following sections will go deeper into these function calls. The Migration App will present this functionality in an easy to use UI, so a regular user will not have to deal with these function calls directly. We will however dive into them in the following sections to dissect how migration works, and outline the process for power users or partners, who want to carry out migration themselves.

Upgrading Dai

In order to upgrade your Dai to MCD, you must use the [swapSaiToDai](#) function in the migration contract. First you must approve that the migration contract can transfer Sai tokens from your account. Then you are ready to invoke the swap by calling the function specifying the amount of Sai you want to upgrade to Dai. After the function call is mined, the upgraded Dai is sent to the Ethereum address initiating the upgrade. A detailed walk-through using CLI tools to carry out these functions can [be found here](#).

From the user perspective, this function simply upgrades a specified amount of Sai to Dai.

Behind the scenes, deposited Sai tokens are used to create a collective CDP in MCD for all migrating users which Dai is minted from. The migration contract will thus take the Sai tokens from your account, and deposit them into the Sai token adapter, which allows the CDP engine Vat to utilize the tokens for CDP actions. The migration contract will then invoke Vat to lock Sai and issue Dai to the Dai adapter. The migration contract will then exit the Dai tokens from the Dai adapter, which is carried out by invoking a mint function on the Dai token contract which will generate new Dai for the originator's Ethereum address. The reason Sai to Dai migration is carried out using the CDP core (vat) of the new system, is that this is the only

component that has the authority to mint new Dai. The process and function calls are outlined in the diagram below.

The following diagram outlines what happens when migrating 10 Sai to 10 Dai.

?

?

Swapping back to Sai

The migration contract also allows users to “go back” by swapping Dai for Sai, using the function [swapDaiToSai](#). In this case, the CDP operation is reversed, as Dai is paid back to the system and Sai is released, just like the repayment of a normal CDP, except with no stability fee cost.

However, this operation requires a surplus of Sai already deposited in the migration contract. Therefore there must be at least an equivalent amount of Sai deposited in the contract, to the amount of Dai you want to redeem.

This function call is very similar to the former, except this time Dai is deposited to the CDP, and Sai collateral released. This requires you to approve that the migration contract can transfer Dai from your wallet, and then you invoke the [swapDaiToSai](#) function with the specified amount of Dai you want to redeem. You can check out [this guide](#) for a more detailed look into how you call the functions.

Migration of CDP

The migration contract also allows users to migrate their CDPs from the SCD core to the MCD core. This is done through the function [migrate](#). The function essentially tries to close your CDP, using excess Sai deposited in the migration contract (by other users who have been upgrading Sai to Dai) to pay your outstanding CDP debt. In order to do so, you need to transfer the control of the CDP to the migration contract. The migration contract will then pay back the debt using Sai deposited in the contract, redeem the ETH collateral, create a new CDP in the MCD system, lock in the ETH collateral, and payback the debt using the generated Dai, resulting in an equivalent CDP debt in MCD.

However, in order to close down the CDP, a stability fee in MKR must be paid, so you need to grant the migration contract approval to spend MKR from your account to carry out the migration.

The migration contract uses a proxy contract to carry out all the above steps in one go. Consequently, the contract can also only be used for CDPs that have been created through a Maker proxy contract. This happens automatically if you have opened your CDP at [cdp.makerdao.com](#). Therefore, you must utilize the [MigrationProxyActions.sol](#) contract to carry out the [migrate function call](#).

If you have created CDPs using third party services that do not use Maker proxies to interact with the CDP core, the migration contract might not work. Instead, you can perform your own manual migration by simply closing down your SCD CDP and moving the ETH to an MCD CDP.

To migrate your CDP, you are also dependant on excess liquidity of Sai in the migration contract to be used to close your CDP. If you have a 10,000 Sai debt CDP you want to migrate, there must be at least 10,000 Sai deposited in the Sai MCD CDP owned by the migration contract (from users upgrading Sai to Dai), to carry out the CDP migration. The migration cannot be carried out partially, why the whole debt of the CDP must be covered by Sai liquidity in the contract to carry out the migration. If you have a large CDP and are concerned about migration, feel free to reach out to the Integrations Team at integrate@makerdao.com

[Read more about the function calls for migrating a CDP here](#)

In Summary

In this guide, we introduced you to the steps of how to upgrade to Multi-Collateral Dai. We have provided you with guidelines for different types of platforms using Dai and for regular Dai holders. As we approach the launch of Multi-Collateral Dai, more details will be made available.

Troubleshooting

If you encounter any issues with your upgrade process, don't hesitate to contact us.

- Contact integrations team - integrate@makerdao.com
- Rocket chat - #dev channel
-

Next Steps

After finishing this guide we think you'll enjoy these next guides:

- Learn about our progress towards the launch of [MCD](#)

- .
-

Resources

Information:

- [Blog post: The Road To Mainnet Release](#)
-

Other Guides:

- [Introduction and Overview of Multi-Collateral Dai: MCD101](#)
- [Using MCD-CLI to create and close a MCD CDP on Kovan](#)
- [Using Seth to create and close an MCD CDP on Kovan](#)
- [Using Seth for MCD migration](#)
- [Add a new collateral type to DCS - Kovan](#)
-

Source code/wiki:

- [Multi Collateral Dai code + wiki](#)
-

[Previous SCD <> MCD Migration](#) Last updated 4 years ago On this page * [Summary](#) * [Introduction](#) * [Important note on naming conventions](#) * [Learning Objective](#) * [Pre-requisites](#) * [Sections](#) * [User and Partner Migration Scenarios](#) * [As a Sai Holder](#) * [As a SCD CDP Owner](#) * [As a Centralized Exchange or Custodial Wallet](#) * [As a Decentralized Exchange](#) * [As a Non-Custodial Wallet](#) * [As a Keeper](#) * [As a Market Maker](#) * [As a CDP Integrator](#) * [Upgrading your CDP integration implementation](#) * [Direct integration with smart contracts](#) * [As a Lending Protocol](#) * [As a Dapp](#) * [As another partner type not mentioned above](#) * [Migration App](#) * [Migration Contract](#) * [Upgrading Dai](#) * [Swapping back to Sai](#) * [Migration of CDP](#) * [In Summary](#)

[Export as PDF](#)