TL;DR

We are happy to present the ParaSwapDebtSwapAdapter

smart contract adapter to the community, allowing users of Aave to swap any debt of a certain asset for variable debt of another asset on all Aave V2 and V3.0.1+ pools

https://github.com/bgd-labs/aave-debt-swap

Context

The Aave v2 and v3 repositories contain smart contracts providing the functionality to both swap one collateral for another (Swap Collateral) and use your collateral to repay your debt (Repay with Collateral).

These contracts are quite general, mainly managing the integration with the interface/features of an external exchange (in this case Paraswap), with specific validations and usage boiling down to the client libraries/UI integrating them. Currently, app.aave.com integrates them.

With the new ParawapDebtSwapAdapter

, we inherit the battle-proven security of the aforementioned adapters to enable a new but quite important feature: allowing users of Aave to exit/arbitrage specific assets on their borrow position.

How does it work?

Let's assume you have an 8000 BUSD

variable debt borrowings and 6 ETH

collateral.

With 160% APY, this is quite an expensive position and you might want to exit it but lack the funds to repay it.

Currently, you have the following ways to exit this position.

1. You can use the "Repay with Collateral" feature to repay the debt with your ETH

collateral.

This has the drawback that you lose exposure to ETH

which might be unintended.

1. When your HF (Health Factor) allows it, you could borrow USDC

→ swap to BUSD on some DEX (Decentralized Exchange) and repay.

This flow involves multiple transactions and it might not be possible to swap your debt in a single run due to HF constraints.

With the ParaSwapDebtSwapAdapter

we provide a 3rd way that allows you to swap your debt in a single transaction without touching your collateral.

To archive this the following steps are performed:

1. Fetch a Paraswap exact out swap route to answer the question of: how much USDC do I need to cover exactly 8000 BUSD debt?

[

swap-paraswap

401×581 26.8 KB

](https://europe1.discourse-cdn.com/business20/uploads/aave/original/2X/6/6b926e619225daf48d329699cc18bf3d97265123.png)

Quote from the Paraswap UI

1. Create a variable borrow mode USDC flash loan with 8000.36 USDC

2. some margin to account for slippage.
3. Swap the USDC

for BUSD

1. Repay the 8000 BUSD

debt

1. Use the potential margin USDC

excess to repay USDC

debt.

And that’s it, now you successfully replaced the BUSD

debt with USDC

debt

Security considerations

Similar to other smart contracts, this type of adapter inherits important properties of base existing infrastructure, already audited and running in production, with the extra logic being dependant on the Aave v2 and v3 security.

These are the considerations we took during the design, development and review process:

- This contract is an extra layer on top of BaseParaswapBuyAdapter which is used in production for ParaSwapRepayAdapter. It uses the exact same mechanism for exact out swap.
- In contrast to ParaSwapRepayAdapter the ParaSwapDebtSwapAdapter will always repay the Aave pool on behalf of the user, which means it is the adapter who does the repayment all the time. So instead of having approvals per transaction and user, the adapter will approve type(uint256).max

once to reduce gas consumption.

- The Aave POOL

is considered a trustable entity for allowance purposes.

- The contract only interacts with msg.sender

and therefore ensures isolation between users.

- The contract is not upgradeable.
- The contract is ownable and will be owned by governance, so the governance will be the only entity able to call tokenRescue

, covering the case of somebody sending tokens by mistake to the adapter.

- The approach with credit delegation and borrow-mode Aave flash loan is very similar to what is done onV2-V3 Migration helper.
- The contract inherits the security and limitations of Aave v2/v3. The contract itself does not validate for frozen/inactive reserves and also does not consider isolation/eMode or borrowCaps. It is the responsibility of the interface integrating this contract to correctly handle all user position compositions and pool configurations, as it is simply too complex to add to the smart contracts layer.
- The contract implements an upper bound of 30% price impact, which would revert any swap.

But again and really importantly, it should be the responsibility of the UI to properly show/warn about the slippage provided by the integrated DEX

.

This slippage has to be properly configured and incorporated into the DebtSwapParams.maxNewDebt

parameter received by the adapter contract.

Next steps

- We have made the code public, so entities with user interfaces supporting Aave can start with the integration. In the case of app.aave.com from @AaveLabs, the existing infrastructure for Repay with Collateral, Collateral Swap, and the v2→v3 migration we helped with, should make the integration straightforward.
- Even if currently this adapter is only covering Paraswap to inherit the existing security assumptions, we encourage other aggregators/DEXes to submit PRs on our repository, integrating the specifics of their platform.