

Update Low-TVL cToken Implementations

Updated as of block [19334007](#) at 2/29/2024, 10:17:23 AM ET

- ID: 96
- Proposer: [0x589CDCf60aea6B961720214e80b713eB66B89A4d](#)
- Start Block: 14503398 (4/1/2022, 7:43:18 PM ET)
- End Block: 14523108 (4/4/2022, 9:55:05 PM ET)
- Targets: [0x041171993284df560249B57358F931D9eB7b925D](#) ;
[0x4B0181102A0112A2ef11AbEE5563bb4a3176c9d7](#) ;
[0x80a2AE356fc9ef4305676f7a3E2Ed04e12C33946](#) ;
[0xe65cdB6479BaC1e22340E4E755fAE7E509EcD06c](#) ;
[0x7713DD9Ca933848F6819F38B8352D9A15EA73F67](#)

Forum Post

Forum post is present here: [Forum Post](#)

Table of contents

- [Proposal Text](#)
- [Checks](#)
 - [Checks Compound Proposal Details](#)  Passed

Proposal Text

Update Low-TVL cToken Implementations

Summary

This proposal is a patch developed by Equilibria which updates the cToken implementation contract for 5 low TVL assets to a more recent Solidity version (0.8.10) and removes error codes in favor of reverts with custom errors. No other core logic changes are intended. The new implementation has been deployed to [0x3363BAe2Fc44dA742Df13CD3ee94b6bB868ea376](#).

Changelog

These changes implement the following:

- Upgrade the Solidity version of the cToken and related contracts to 0.8.10 - all contracts in the repo were changed to 0.8.10 but only the cTokens will be upgraded as part of upcoming governance proposals. This is due to the complexity of having multiple Solidity versions in the same repo.
- Remove the usage of SafeMath and CarefulMath in favor of Solidity 0.8's checked math - Solidity will now automatically revert when math errors occur (overflows, division by zero, etc)
- Remove the custom errorCode return values in favor of reverts and custom errors - this allows for a more structured way to deal with errors rather than enum or string comparisons.

It is important to note that the goal is to have no behavior changes in the happy path case, and to only move away from errorCodes and to revert in the failure case (both math errors and checks). All existing unit and scenario tests should pass with only changes to the error code cases.

More cTokens will be upgraded in a future proposal if this one passes and causes no issues.

Development






The code changes can be viewed here: [Pull Request #152](#).

Audits of these changes were completed by ChainSecurity and OpenZeppelin and all issues were either fixed or out of scope for this change. For further discussion, please view the [Community Forum thread](#).

Checks

Checks Compound Proposal Details Passed

Info:

- 1-  Function `_setImplementation(address,bool,bytes)` is removed from `0x041171993284df560249b57358f931d9eb7b925d` contract
- 2-  Function `_setImplementation(address,bool,bytes)` is removed from `0x4b0181102a0112a2ef11abee5563bb4a3176c9d7` contract
- 3-  Function `_setImplementation(address,bool,bytes)` is removed from `0x80a2ae356fc9ef4305676f7a3e2ed04e12c33946` contract
- 4-  Function `_setImplementation(address,bool,bytes)` is removed from `0xe65cdb6479bac1e22340e4e755fae7e509ecd06c` contract
- 5-  Function `_setImplementation(address,bool,bytes)` is removed from `0x7713dd9ca933848f6819f38b8352d9a15ea73f67` contract