

# Update High-TVL cToken Implementations

Updated as of block [19334254](#) at 2/29/2024, 11:07:11 AM ET

- ID: 104
- Proposer: [0x589CDCf60aea6B961720214e80b713eB66B89A4d](#)
- Start Block: 14700539 (5/2/2022, 5:00:35 PM ET)
- End Block: 14720249 (5/5/2022, 7:58:23 PM ET)
- Targets: [0xf650C3d88D12dB855b8bf7D11Be6C55A4e07dCC9](#) ;  
[0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643](#) ;  
[0xccF4429DB6322D5C611ee964527D42E5d685DD6a](#) ;  
[0x3d9819210A31b4961b30EF54bE2aeD79B9c9Cd3B](#) ;  
[0x39AA39c021dfbaE8faC545936693aC917d5E7563](#) ;  
[0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48](#)

## Forum Post

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

## Table of contents

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

## Proposal Text

### Update High-TVL cToken Implementations

#### Summary

This proposal is a patch developed by Equilibria which updates the cToken implementation contract for high TVL assets. The new implementation has been deployed to [0x3363BAe2Fc44dA742Df13CD3ee94b6bB868ea376](#).

This is the final proposal to bring all upgradeable cTokens up to the new implementation. In addition to updating the implementation, it grants \$350,000[1] worth of COMP to Equilibria for the upgrade work and 79,764.36 USDC to Compound Labs for the ChainSecurity audit of the protocol[2].

[1] COMP value based on the current spot price at time of this proposal (\$114) [2] ChainSecurity Payment Transaction can be viewed [here](#)

#### 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.

The previous upgrade can be seen here: [Proposal 101](#)

## 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 `0xf650c3d88d12db855b8bf7d11be6c55a4e07dcc9` contract
- 2-  Function `_setImplementation(address,bool,bytes)` is removed from `0x5d3a536e4d6dbd6114cc1ead35777bab948e3643` contract
- 3-  Function `_setImplementation(address,bool,bytes)` is removed from `0xccf4429db6322d5c611ee964527d42e5d685dd6a` contract
- 4-  Grant **3,070.1754385965** [COMP](#) tokens to [0x589CDCf60aea6B961720214e80b713eB66B89A4d](#).
- 5- Reduce reserves of [cUSDC](#) by 79,764.36 [USDC](#). Remaining total reserves would be 91,220.03548221
- 6-  Transfer **79,764.36** [USDC](#) to [0xe8F0c9059b8Db5B863d48dB8e8C1A09f97D3B991](#).