
Security Review Report

NM-0475 ORIGAMI



NETHERMIND
SECURITY

(April 04, 2025)

Contents

1	Executive Summary	2
2	Audited Files	3
3	Summary of Issues	3
4	System Overview	4
4.1	The Vault and Manager Contracts	5
4.1.1	OrigamiTokenizedBalanceSheetVault	5
4.1.2	OrigamiHOmVault	6
4.1.3	OrigamiHOmManager	6
5	Risk Rating Methodology	8
6	Issues	9
6.1	[Best Practices] The Vault allows joins with zero shares	9
7	Documentation Evaluation	10
8	Test Suite Evaluation	11
8.1	Compilation Output	11
8.2	Tests Output	12
8.3	Automated Tools	22
8.3.1	AuditAgent	22
9	About Nethermind	23

1 Executive Summary

This document presents the security review performed by [Nethermind Security](#) for [Origami protocol](#) smart contracts. Origami is a decentralized finance protocol designed to offer users leveraged exposure to the Olympus ecosystem through a tokenized vault system.

The system uses a tokenized balance sheet to manage assets and liabilities, enabling accurate share pricing and redemptions. It includes automated LTV optimization, surplus management for liquidity, and vote delegation for governance.

Users can deposit gOHM tokens that will be used as collateral in MonoCooler to borrow USDS that may be used to do buybacks and increase the gOHM entitled to users. The Vault automates a strategy allowing users to leverage their gOHM to earn more gOHM.

The Vault shares (HOHm) can also be bridged through layer zero to be traded and used on other chains.

The audited code comprises 1,582 lines of Solidity code. **The audit was performed using** (a) manual analysis of the codebase, (b) automated analysis tools, and (c) creation of test cases. **Along this document, we report** one points of attention, classified as **Best Practices**, as shown in Fig. 1.

This document is organized as follows. Section 2 presents the files in the scope. Section 3 summarizes the issues. Section 4 presents the system overview. Section 5 discusses the risk rating methodology. Section 6 details the issues. Section 7 discusses the documentation provided by the client for this audit. Section 8 presents the compilation, tests, and automated tests. Section 9 concludes the document.

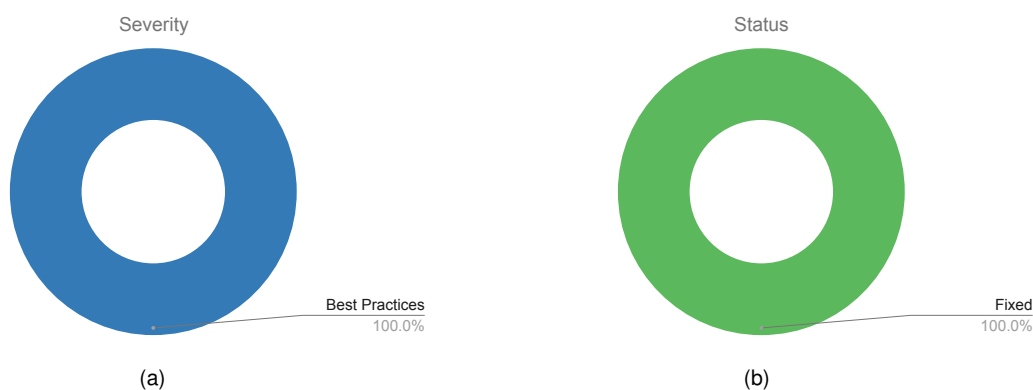


Fig. 1: Distribution of issues: Critical (0), High (0), Medium (0), Low (0), Undetermined (0), Informational (0), Best Practices (1).
Distribution of status: Fixed (1), Acknowledged (0), Mitigated (0), Unresolved (0)

Summary of the Audit

Audit Type	Security Review
Initial Report	April 04, 2025
Response from Client	Regular responses during audit engagement
Final Report	April 04, 2025
Repository	origami
Start Commit	6b0e28eb43cd32f0bf61c600d9c9e561df6796de
Final Commit	3fb41ec2995a8bac04cdef332d21ec4348ed7572
Documentation	Origami Spec and Origami Doc Repo
Documentation Assessment	High
Test Suite Assessment	High

2 Audited Files

	Contract	LoC	Comments	Ratio	Blank	Total
1	libraries/OlympusCoolerDelegation.sol	142	55	38.7%	22	219
2	common/OrigamiTokenizedBalanceSheetVault.sol	591	192	32.5%	108	891
3	common/swappers/OrigamiSwapperWithCallback.sol	39	23	59.0%	13	75
4	common/access/OrigamiElevatedAccess.sol	7	5	71.4%	2	14
5	common/access/Whitelisted.sol	23	27	117.4%	9	59
6	common/access/OrigamiOfElevatedAccess.sol	13	6	46.2%	3	22
7	common/access/OrigamiElevatedAccessUpgradeable.sol	12	10	83.3%	4	26
8	common/access/OrigamiElevatedAccessBase.sol	49	37	75.5%	14	100
9	common/omnichain/OrigamiTokenTeleporter.sol	17	7	41.2%	2	26
10	common/omnichain/OrigamiTeleportableToken.sol	65	19	29.2%	11	95
11	common/omnichain/OrigamiOFT.sol	32	7	21.9%	5	44
12	investments/olympus/OrigamiHOHmManager.sol	410	146	35.6%	96	652
13	investments/olympus/OrigamiHOHmVault.sol	182	48	26.4%	35	265
	Total	1582	582	36.8%	324	2488

3 Summary of Issues

	Finding	Severity	Update
1	The Vault allows joins with zero shares	Best Practices	Fixed

4 System Overview

This review covers 13 contracts organized into four main components: access control, an OFT (Omnichain Fungible Token) standard token, a tokenized vault based on the ERC-4626 standard, and a manager contract responsible for collateral management and enforcing maximum borrowing limits in the Olympus MonoCooler system. Fig. 2 presents a package diagram that highlights the main packages and contracts included in the scope of this review.

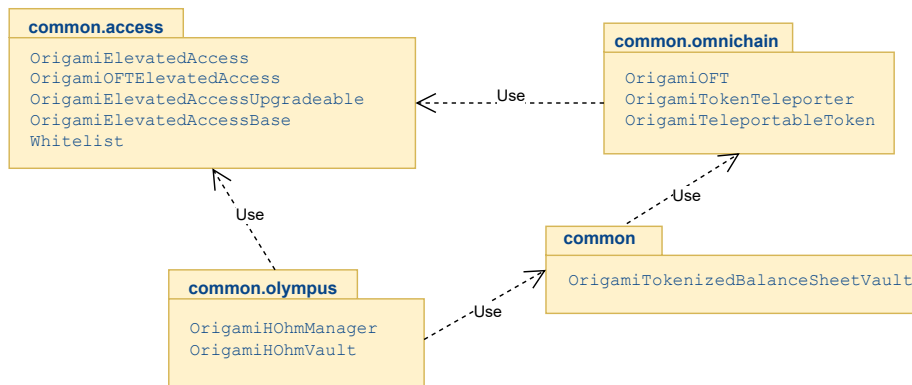


Fig. 2: The package diagram shows the main packages and contracts for a system overview.

Origami system relies on LayerZero to use OFT (Omnichain Fungible Token) standard to allow hOHM to be transferred cross-chain. As presented in Fig. 3, the `OrigamiTeleportableToken` contract is an ERC20 token with the support of the `ERC20Permit` that does not require token approval to be spent by the trusted teleporter. The `OrigamiOFT` contract extends the LayerZero OFT contract, allowing cross-chain token transfers while preserving compatibility with ERC-20 and its extensions. All these contracts also inherit from `OrigamiElevatedAccess` to manage administrative permissions.

Fig. 3 shows that `OrigamiTokenTeleporter` contract extends the `OFTAdapter` to enable interoperability of hOHM across multiple chains. Since this contract is deployed on the mainnet, the inner token is locked, rather than burned, when transferred through it.

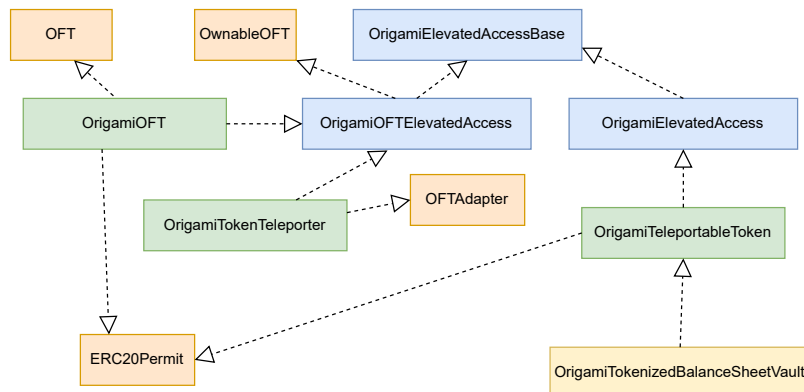


Fig. 3: The class diagram highlights the omnichain token.

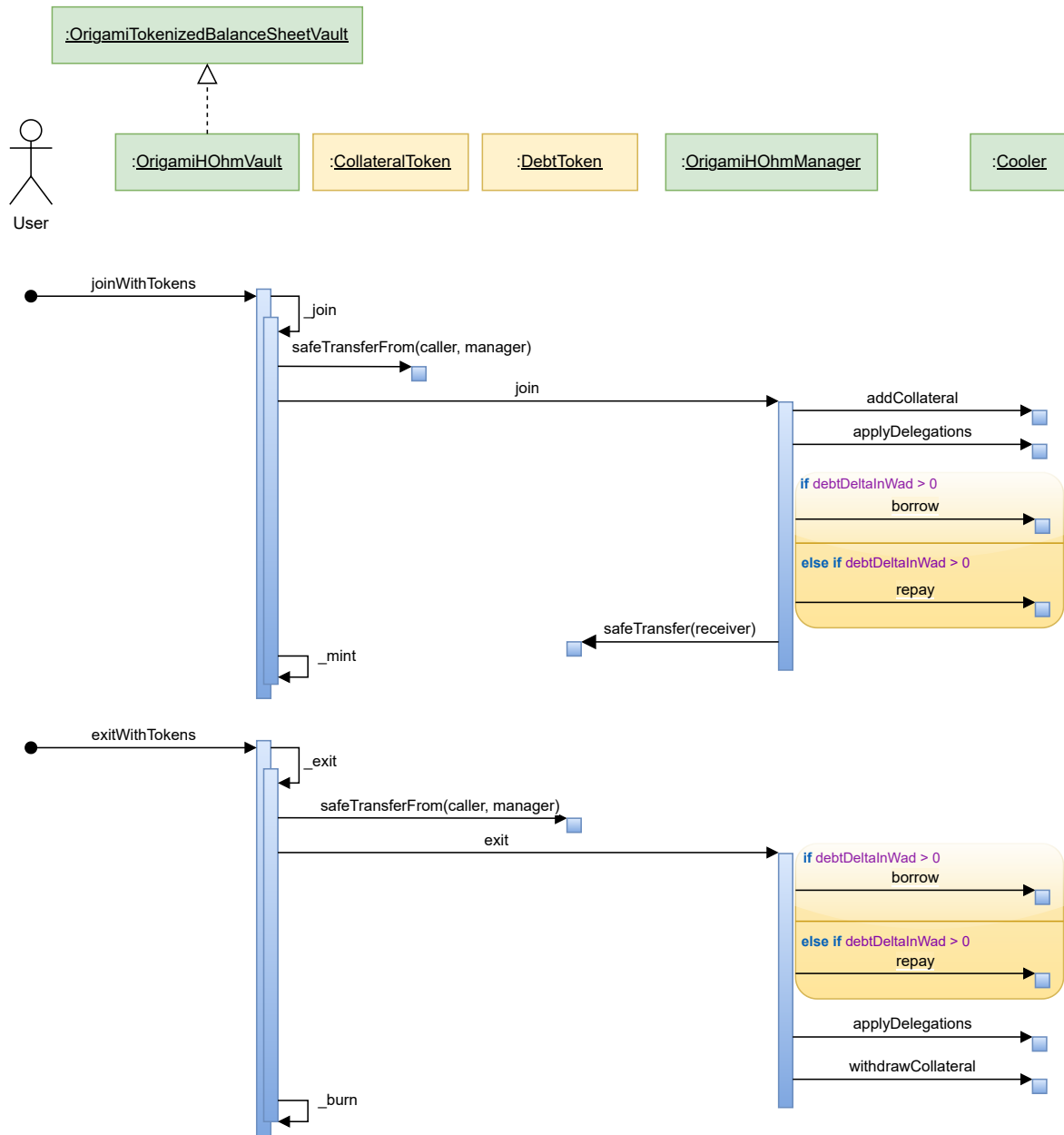


Fig. 4: The sequence diagram illustrates the main function calls involved in joining and exiting the vault. For simplicity, it focuses on the token-based join and exit operations. However, the mechanism also supports joining and exiting using shares.

4.1 The Vault and Manager Contracts

The contracts `OrigamiHOHmVault` and `OrigamiHOHmManager` work together to implement a tokenized financial mechanism based on collateral and debt. The Vault serves as a tokenized balance sheet vault, allowing users to mint and burn shares in exchange for assets (collateral) and liabilities (debt). The Manager contract handles interactions with the Olympus MonoCooler system to add/remove collateral and to borrow or repay debt, always aiming to maintain the Loan-to-Value (LTV) ratio at the maximum allowed by Cooler. The Manager also manages the delegation of gOHM voting power and handles surplus debt token strategies, optimizing for yield and system stability. Fig. 4 illustrates this interaction between both contracts.

4.1.1 OrigamiTokenizedBalanceSheetVault

As presented in Fig 3, the `OrigamiTokenizedBalanceSheetVault` contract extends the `OrigamiTeleportableToken`. Basically, the shares in this vault represent a proportional part of both the assets and liabilities within this structure.

- **Assets:** Represent the positive value held by the vault. They may consist of zero or more ERC20 tokens, with each asset having an independent balance.
- **Liabilities:** Represent the debt owed by the vault. Like assets, these may include zero or more ERC20 tokens with varying balances.

Minting Shares. When users mint new shares:

- They provide a proportional amount of each **asset** based on the current balance sheet.
- They receive a proportional amount of each liability.
- A number of shares is minted to represent the proportional ownership of the vault balance sheet equity.

Redeeming Shares. Users can redeem their shares:

- They provide a proportional amount of liability tokens for the respective number of shares.
- They receive a proportional amount of each asset according to the respective number of shares.
- Then, the corresponding shares are burned.

Dynamic Balance Sheet:

- The asset and liability token *balances* can change over time due to yield generation, rebalancing, or debt accrual.
- The asset and liability token *addresses* can also change (e.g., rolling yield tokens or removal of tokens with zero balance).

Core functions. Fig. 4 illustrates a sequence diagram for two of the core functions. The contract exposes several public and external functions. Inspired by ERC4626, it provides the following core operations:

- `seed(assetAmounts, liabilityAmounts, sharesToMint, receiver, newMaxTotalSupply)`: It is invoked by an elevated access role to initialize the vault. The caller must provide the amount of each asset and liability, along with the number of shares to mint. These values define the initial share price for each asset and liability.
- `joinWithToken(tokenAddress, amount)`: Mints shares based on the input of a single asset or liability. The vault derives the share amount and any remaining required token transfers proportionally. Fig. 2 depicts how this function interacts with other contracts.
- `joinWithShares(shares)`: Mints a specified number of shares. The vault pulls a proportional amount of assets from the user and sends the corresponding liabilities. The flow is similar to the `joinWithToken` function, the difference is based on the input parameter.
- `exitWithToken(tokenAddress, amount)`: Burns shares to allow for redemption based on a single token (asset or liability). The vault determines the proportional share amount and additional transfers. Fig. 2 depicts how this function interacts with other contracts.
- `exitWithShares(shares)`: Burns a specified number of shares. The vault sends a proportional amount of assets to the user and pulls the corresponding liabilities. The flow is similar to the `joinWithToken` function, the difference is based on the input parameter.

4.1.2 OrigamiHOHmVault

The `OrigamiHOHmVault` contract is a specialized implementation of the `OrigamiTokenizedBalanceSheetVault` (shown in Fig. 4), designed to manage a leveraged gOHM position using the MonoCooler lending system. It allows users to deposit gOHM as collateral and receive a combination of vault shares (hOHM) and borrowed debt tokens (e.g., USDS).

This vault tracks a single asset (the gOHM collateral) and a single liability (the debt token). While the base vault handles share minting, burning, and conversions between tokens and shares, `OrigamiHOHmVault` adds logic specific to the Olympus ecosystem. It delegates all borrowing and collateral interactions to the `OrigamiHOHmManager` contract.

The key responsibilities of `OrigamiHOHmVault` include:

- Defining the tokenized asset and liability types (gOHM and debt token).
- Exposing vault controls like `areJoinsPaused`, `areExitsPaused`, and `burn`.
- Synchronizing the delegation of voting power in gOHM.
- Triggering interactions with the manager when users join or exit the vault.
- Managing user share balances and ensuring that delegation amounts reflect current gOHM holdings after minting, transfers, or redemptions.

4.1.3 OrigamiHOHmManager

The `OrigamiHOHmManager` contract serves as the core logic handler for interactions between the `OrigamiHOHmVault` and Olympus MonoCooler lending platform. Its primary role is to abstract and automate the process of managing collateral and debt within the vault.

Responsibilities: This manager contract is responsible for:

- Adding and removing gOHM collateral to/from the `MonoCooler` contract.
- Managing borrowing and repay operations in a way that preserves a target Loan-To-Value (LTV) ratio.

- Automatic synchronization of gOHM delegation through the Olympus delegation module (DLGTEV1).
- Handling exit fees and performance fees during vault exits.
- Maintain a surplus buffer in debtToken (e.g., USDS) to ensure liquidity and support redemptions.
- Interfacing with a Swapper contract to periodically perform buybacks and burn hOHM using excess debt tokens.

Delegation Management: The manager tracks the delegation information per account and updates the delegation amounts automatically when the vault shares change. It computes a user's proportional collateral based on their share of the total supply and submits the corresponding delegation update to the MonoCooler contract.

Debt Optimization: Upon each vault join or exit, the OrigamiHOhmManager invokes the `_coolerMaxBorrow` function to determine the optimal change in debt necessary to maintain the maximum allowed origination LTV. The result is applied by calling `borrow` or `repay` on MonoCooler. This flow is presented in Fig. 4.

Surplus strategy: Any surplus debtToken held by the contract can be deposited in a yield-bearing savings vault (e.g., sUSDS) which has a higher interest rate than the cooler interest rate. The contract also provides the `sweep` function that can be called to use the surplus token to buy hOHM from the market and then burn it. The goal for this function is to decrease the total supply, consequently, increase the share price of both the collateral and the debt tokens per hOHM.

5 Risk Rating Methodology

The risk rating methodology used by [Nethermind Security](#) follows the principles established by the [OWASP Foundation](#). The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely the finding is to be uncovered and exploited by an attacker. This factor will be one of the following values:

- a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;
- b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;
- c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

- a) **High**: The issue can cause significant damage, such as loss of funds or the protocol entering an unrecoverable state;
- b) **Medium**: The issue can cause moderate damage, such as impacts that only affect a small group of users or only a particular part of the protocol;
- c) **Low**: The issue can cause little to no damage, such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding, other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Severity Risk		
Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Info/Best Practices	Low	Medium
	Undetermined	Undetermined	Undetermined	Undetermined
		Low	Medium	High
		Likelihood		

To address issues that do not fit a High/Medium/Low severity, [Nethermind Security](#) also uses three more finding severities: **Informational**, **Best Practices**, and **Undetermined**.

- a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to pass to the client formally;
- b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;
- c) **Undetermined** findings are used when we cannot predict the impact or likelihood of the issue.

6 Issues

6.1 [Best Practices] The Vault allows joins with zero shares

File(s): [apps/protocol/contracts/common/OrigamiTokenizedBalanceSheetVault.sol](#)

Description: The Vault allows users to execute join operations that would result in zero shares due to rounding. Users could join but not receive any shares in exchange.

Recommendation(s): Consider reverting when minting zero shares.

Status: Fixed.

Update from the client: Fixed in commit [3fb41e](#).

7 Documentation Evaluation

Software documentation refers to the written or visual information that describes the functionality, architecture, design, and implementation of software. It provides a comprehensive overview of the software system and helps users, developers, and stakeholders understand how the software works, how to use it, and how to maintain it. Software documentation can take different forms, such as user manuals, system manuals, technical specifications, requirements documents, design documents, and code comments. Software documentation is critical in software development, enabling effective communication between developers, testers, users, and other stakeholders. It helps to ensure that everyone involved in the development process has a shared understanding of the software system and its functionality. Moreover, software documentation can improve software maintenance by providing a clear and complete understanding of the software system, making it easier for developers to maintain, modify, and update the software over time. Smart contracts can use various types of software documentation. Some of the most common types include:

- Technical whitepaper: A technical whitepaper is a comprehensive document describing the smart contract's design and technical details. It includes information about the purpose of the contract, its architecture, its components, and how they interact with each other;
- User manual: A user manual is a document that provides information about how to use the smart contract. It includes step-by-step instructions on how to perform various tasks and explains the different features and functionalities of the contract;
- Code documentation: Code documentation is a document that provides details about the code of the smart contract. It includes information about the functions, variables, and classes used in the code, as well as explanations of how they work;
- API documentation: API documentation is a document that provides information about the API (Application Programming Interface) of the smart contract. It includes details about the methods, parameters, and responses that can be used to interact with the contract;
- Testing documentation: Testing documentation is a document that provides information about how the smart contract was tested. It includes details about the test cases that were used, the results of the tests, and any issues that were identified during testing;
- Audit documentation: Audit documentation includes reports, notes, and other materials related to the security audit of the smart contract. This type of documentation is critical in ensuring that the smart contract is secure and free from vulnerabilities.

These types of documentation are essential for smart contract development and maintenance. They help ensure that the contract is properly designed, implemented, and tested, and they provide a reference for developers who need to modify or maintain the contract in the future.

Remarks about the Origami documentation

The Origami team has provided comprehensive documentation that describes expected flows and behaviors, including detailed examples of minting, redeeming, buybacks, and seeding operations. The codebase is also well-commented, addressing both business logic and low-level implementation details. Moreover, the team addressed all questions and concerns raised by the Nethermind Security team, offering valuable insights and demonstrating a strong understanding of the project's technical aspects.

8 Test Suite Evaluation

8.1 Compilation Output

```
1 forge build --skip test
2 [] Compiling...
3 [] Compiling 409 files with Solc 0.8.22
4 [] Solc 0.8.22 finished in 31.37s
5 Compiler run successful!
```

8.2 Tests Output

```
forge test --match-path "**/{olympus,tokenizedBalanceSheet}/**"
[] Compiling...
No files changed, compilation skipped

Ran 2 tests for
  ↳ ../OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestViewsReplicateMaxExitIssue
[PASS] test_maxExit_nearEmpty_asset1() (gas: 265875)
[PASS] test_maxExit_nearEmpty_debt2() (gas: 267023)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 27.54ms (2.38ms CPU time)

Ran 2 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestDebtTokenChange
[PASS] test_changeDebtToken_noSurplus() (gas: 3350782)
[PASS] test_changeDebtToken_withSurplus() (gas: 3419204)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 33.11ms (7.61ms CPU time)

Ran 9 tests for
  ↳ ../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAdminWithFees
[PASS] test_initialization() (gas: 225329)
[PASS] test_recoverToken_assetsAndLiabilities() (gas: 777589)
[PASS] test_recoverToken_success() (gas: 2039009)
[PASS] test_seedDeposit_failure_badParams() (gas: 13867630)
[PASS] test_seedDeposit_failure_maxTooLow() (gas: 13861501)
[PASS] test_seedDeposit_success() (gas: 14417189)
[PASS] test_seed_failure_alreadySeeded() (gas: 18673)
[PASS] test_setMaxTotalSupply_failure_zeroSupply() (gas: 4461507)
[PASS] test_setMaxTotalSupply_success() (gas: 23664)
Suite result: ok. 9 passed; 0 failed; 0 skipped; finished in 33.59ms (8.44ms CPU time)

Ran 10 tests for
  ↳ ../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestViewsWithFees
[PASS] test_assetTokens() (gas: 21421)
[PASS] test_availableSharesCapacity() (gas: 29970)
[PASS] test_isBalanceSheetToken() (gas: 49905)
[PASS] test_liabilityTokens() (gas: 21423)
[PASS] test_maxExit() (gas: 252556)
[PASS] test_maxJoin() (gas: 193731)
[PASS] test_previewExit() (gas: 236131)
[PASS] test_previewJoin() (gas: 236068)
[PASS] test_supportsInterface() (gas: 18845)
[PASS] test_tokens() (gas: 34938)
Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 6.26ms (3.01ms CPU time)

Ran 12 tests for ../OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestJoinWithShares
[PASS] test_joinWithShares_afterShareIncrease() (gas: 1085848)
[PASS] test_joinWithShares_badReceiver() (gas: 524006)
[PASS] test_joinWithShares_basic() (gas: 647370)
[PASS] test_joinWithShares_beforeShareIncrease_asset() (gas: 1073119)
[PASS] test_joinWithShares_differentReceiver() (gas: 652232)
[PASS] test_joinWithShares_exactlyOneShare() (gas: 647543)
[PASS] test_joinWithShares_fail_tooMuch() (gas: 399616)
[PASS] test_joinWithShares_multiple() (gas: 953093)
[PASS] test_joinWithShares_noAssetLeft() (gas: 626167)
[PASS] test_joinWithShares_noLiabilitiesLeft() (gas: 641171)
[PASS] test_joinWithShares_partialShare() (gas: 662579)
[PASS] test_joinWithShares_zeroAmount() (gas: 512392)
Suite result: ok. 12 passed; 0 failed; 0 skipped; finished in 37.60ms (12.46ms CPU time)

Ran 9 tests for
  ↳ ../DynamicAssetsTokenizedBalanceSheetVault.t.sol:DynamicOrigamiTokenizedBalanceSheetVaultTestRolloverJoinWithShares
[PASS] test_joinWithShares_basic_fullRebalance() (gas: 5234693)
[PASS] test_joinWithShares_basic_randomRebalance() (gas: 5461576)
[PASS] test_joinWithShares_multiple_fullRebalance() (gas: 5390765)
[PASS] test_joinWithShares_multiple_rebalance_joinWithShares_multiple() (gas: 5820562)
[PASS] test_joinWithShares_partialShare_fullRebalance() (gas: 5234735)
[PASS] test_joinWithShares_partialShare_randomRebalance() (gas: 5461553)
[PASS] test_joinWithToken_before_full_rebalance_exitWithShares_after_emptyies_vault() (gas: 5416953)
[PASS] test_joinWithToken_rebalance_shareIncrease_emptyies_vault_withShares() (gas: 5513288)
[PASS] test_joinWithToken_singleToken_rollover_previewJoinWithShares() (gas: 2941497)
Suite result: ok. 9 passed; 0 failed; 0 skipped; finished in 47.29ms (21.71ms CPU time)
```

```

1  Ran 12 tests for .../OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestExitWithShares
2  [PASS] test_exitWithShares_afterShareIncrease() (gas: 1279639)
3  [PASS] test_exitWithShares_badReceiver() (gas: 664646)
4  [PASS] test_exitWithShares_basic() (gas: 853419)
5  [PASS] test_exitWithShares_beforeShareIncrease_asset() (gas: 1279525)
6  [PASS] test_exitWithShares_differentReceiver() (gas: 869335)
7  [PASS] test_exitWithShares_fail_exitTooMuch() (gas: 661208)
8  [PASS] test_exitWithShares_multiple() (gas: 998146)
9  [PASS] test_exitWithShares_noAssetLeft() (gas: 859438)
10 [PASS] test_exitWithShares_noLiabilitiesLeft() (gas: 842383)
11 [PASS] test_exitWithShares_onBehalfOf() (gas: 1405962)
12 [PASS] test_exitWithShares_zeroAmount() (gas: 754543)
13 [PASS] test_exitWithShares_zeroSharesOwner() (gas: 71328)
14 Suite result: ok. 12 passed; 0 failed; 0 skipped; finished in 20.32ms (17.32ms CPU time)
15
16 Ran 8 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerMaxBorrow
17 [PASS] test_maxBorrowFromCooler_borrowDisabled() (gas: 288743)
18 [PASS] test_maxBorrowFromCooler_borrowWithNoPriorSurplus_notSavings() (gas: 526887)
19 [PASS] test_maxBorrowFromCooler_borrowWithNoPriorSurplus_savings() (gas: 478622)
20 [PASS] test_maxBorrowFromCooler_borrowWithSomePriorSurplus_savings() (gas: 494334)
21 [PASS] test_maxBorrowFromCooler_increasedOriginationLtv() (gas: 605985)
22 [PASS] test_maxBorrowFromCooler_noChange() (gas: 483608)
23 [PASS] test_maxBorrowFromCooler_noPosition() (gas: 33525)
24 [PASS] test_maxBorrowFromCooler_repay() (gas: 714149)
25 Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 8.69ms (4.46ms CPU time)
26
27 Ran 17 tests for
28   ↳ .../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestJoinWithToken
29 [PASS] test_exit_fail_paused() (gas: 971654)
30 [PASS] test_joinWithToken_afterShareIncrease() (gas: 1092343)
31 [PASS] test_joinWithToken_badReceiver() (gas: 526330)
32 [PASS] test_joinWithToken_badTokenAddress_cappedTotalSupply() (gas: 2168638)
33 [PASS] test_joinWithToken_badTokenAddress_unlimitedTotalSupply() (gas: 2180288)
34 [PASS] test_joinWithToken_basic_asset1() (gas: 666648)
35 [PASS] test_joinWithToken_basic_asset2() (gas: 667036)
36 [PASS] test_joinWithToken_basic_debt1() (gas: 667720)
37 [PASS] test_joinWithToken_basic_debt2() (gas: 668092)
38 [PASS] test_joinWithToken_beforeShareIncrease_asset() (gas: 1092300)
39 [PASS] test_joinWithToken_differentReceiver() (gas: 671561)
40 [PASS] test_joinWithToken_fail_tooMuch() (gas: 421684)
41 [PASS] test_joinWithToken_multiple() (gas: 993536)
42 [PASS] test_joinWithToken_noAssetLeft() (gas: 196756)
43 [PASS] test_joinWithToken_noLiabilitiesLeft() (gas: 193676)
44 [PASS] test_joinWithToken_zeroAmount() (gas: 531776)
45 [PASS] test_join_fail_paused() (gas: 442496)
46 Suite result: ok. 17 passed; 0 failed; 0 skipped; finished in 19.17ms (15.86ms CPU time)
47
48 Ran 17 tests for test/foundry/unit/investments/olympus/OrigamiHOhmVault.t.sol:OrigamiHOhmVaultTestDelegations
49 [PASS] test_delegateVotingPower_noUpfrontCollateral() (gas: 733417)
50 [PASS] test_delegateVotingPower_ohmBacking_increaseAndJoinAgain() (gas: 1397383)
51 [PASS] test_delegateVotingPower_ohmBacking_increaseAndReDelegate() (gas: 1114987)
52 [PASS] test_delegateVotingPower_ohmBacking_increaseAndSync() (gas: 1114351)
53 [PASS] test_delegateVotingPower_ohmBacking_increaseThenExit() (gas: 1388468)
54 [PASS] test_delegateVotingPower_other() (gas: 1018335)
55 [PASS] test_delegateVotingPower_reduceOnExit() (gas: 1256765)
56 [PASS] test_delegateVotingPower_removeDelegate() (gas: 1050083)
57 [PASS] test_delegateVotingPower_self() (gas: 1010742)
58 [PASS] test_delegateVotingPower_zeroSupply() (gas: 9105306)
59 [PASS] test_transfer_toOther_noDelegation() (gas: 747860)
60 [PASS] test_transfer_toOther_noDelegationFrom_withDelegationTo() (gas: 1114408)
61 [PASS] test_transfer_toOther_withDelegationFrom_noDelegationTo() (gas: 1092164)
62 [PASS] test_transfer_toOther_withDelegationFrom_withDelegationTo() (gas: 1210383)
63 [PASS] test_transfer_toSelf_noDelegation() (gas: 740279)
64 [PASS] test_transfer_toSelf_withDelegation() (gas: 1059416)
65 [PASS] test_transfer_zeroAmount() (gas: 747761)
66 Suite result: ok. 17 passed; 0 failed; 0 skipped; finished in 27.31ms (23.11ms CPU time)
67
68 Ran 15 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerTestAccess
69 [PASS] test_exit_access() (gas: 16128)
70 [PASS] test_join_access() (gas: 15871)

```

```

1  [PASS] test_recoverToken_access() (gas: 18073)
2  [PASS] test_setCoolerBorrowsDisabled_access() (gas: 15803)
3  [PASS] test_setDebtTokenFromCooler_access() (gas: 17921)
4  [PASS] test_setDelegationAmount1_access() (gas: 15784)
5  [PASS] test_setDelegationAmount2_access() (gas: 16112)
6  [PASS] test_setExitFees_access() (gas: 15819)
7  [PASS] test_setFeeCollector_access() (gas: 17921)
8  [PASS] test_setPerformanceFeesBps_access() (gas: 15775)
9  [PASS] test_setSweepParams_access() (gas: 15874)
10 [PASS] test_setSweepSwapper_access() (gas: 17968)
11 [PASS] test_sweep_access() (gas: 16384)
12 [PASS] test_syncDebtTokenSavings_access() (gas: 15697)
13 [PASS] test_updateDelegateAndAmount_access() (gas: 15943)
14 Suite result: ok. 15 passed; 0 failed; 0 skipped; finished in 6.61ms (916.46µs CPU time)
15
16 Ran 2 tests for
17   ↳ ../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestPermitAndBurn
18 [PASS] test_burn() (gas: 682536)
19 [PASS] test_permit() (gas: 91007)
20 Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 10.02ms (6.17ms CPU time)
21
22 Ran 10 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestJoinAndExit
23 [PASS] test_exitWithShares() (gas: 671923)
24 [PASS] test_exitWithToken_gohm() (gas: 658192)
25 [PASS] test_exitWithToken_other() (gas: 35163)
26 [PASS] test_exitWithToken_usds() (gas: 658776)
27 [PASS] test_exit_fail_paused() (gas: 471625)
28 [PASS] test_joinWithShares() (gas: 427515)
29 [PASS] test_joinWithToken_gohm() (gas: 445295)
30 [PASS] test_joinWithToken_other() (gas: 32102)
31 [PASS] test_joinWithToken_usds() (gas: 446160)
32 [PASS] test_join_fail_paused() (gas: 95224)
33 Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 8.54ms (4.28ms CPU time)
34
35 Ran 24 tests for test/foundry/unit/investments/olympus/OrigamiHOHmManager.t.sol:OrigamiHOHmManagerTestAdmin
36 [PASS] test_constructor_fail_badSavings() (gas: 1714915)
37 [PASS] test_constructor_fail_decimals() (gas: 121464)
38 [PASS] test_constructor_fail_feeBps() (gas: 223131)
39 [PASS] test_constructor_noSavings() (gas: 4029579)
40 [PASS] test_initialization() (gas: 4143124)
41 [PASS] test_recoverToken_fail() (gas: 31108)
42 [PASS] test_recoverToken_success() (gas: 2043237)
43 [PASS] test_setCoolerBorrowsDisabled() (gas: 23065)
44 [PASS] test_setDebtTokenFromCooler_changeDebtToken_noSavings() (gas: 1966156)
45 [PASS] test_setDebtTokenFromCooler_changeDebtToken_withSavings() (gas: 3569820)
46 [PASS] test_setDebtTokenFromCooler_fail_badSavings() (gas: 1624580)
47 [PASS] test_setDebtTokenFromCooler_fail_notPaused() (gas: 73986)
48 [PASS] test_setDebtTokenFromCooler_fromUnsetSavings() (gas: 1680390)
49 [PASS] test_setDebtTokenFromCooler_noChange() (gas: 122999)
50 [PASS] test_setDebtTokenFromCooler_unsetSavings() (gas: 110175)
51 [PASS] test_setExitFees_fail() (gas: 13501)
52 [PASS] test_setExitFees_success() (gas: 21775)
53 [PASS] test_setFeeCollector_fail() (gas: 13686)
54 [PASS] test_setFeeCollector_success() (gas: 24363)
55 [PASS] test_setPerformanceFeesBps_fail() (gas: 13511)
56 [PASS] test_setPerformanceFeesBps_success() (gas: 21731)
57 [PASS] test_setSweepParams() (gas: 28720)
58 [PASS] test_setSweepSwapper_fail() (gas: 13728)
59 [PASS] test_setSweepSwapper_success() (gas: 24409)
60 Suite result: ok. 24 passed; 0 failed; 0 skipped; finished in 8.72ms (3.49ms CPU time)
61
62 Ran 2 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestMulticall
63 [PASS] test_multicall_fail_joinAndTransferTooMuch() (gas: 388602)
64 [PASS] test_multicall_success_joinAndDelegate() (gas: 726255)
65 Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 5.24ms (1.15ms CPU time)
66
67 Ran 5 tests for
68   ↳ ../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestViewsNoFees
69 [PASS] test_maxExit() (gas: 143347)
70 [PASS] test_maxJoin() (gas: 85881)
71 [PASS] test_noFees() (gas: 14973397)
72 [PASS] test_previewExit() (gas: 161012)
73 [PASS] test_previewJoin() (gas: 160965)
74 Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 9.58ms (4.89ms CPU time)

```

```

1  Ran 20 tests for
   ↳ .../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestExitWithToken
2  [PASS] test_exitWithShares_fromExitWithTokenQuote() (gas: 850567)
3  [PASS] test_exitWithShares_fullUnwind() (gas: 1024451)
4  [PASS] test_exitWithToken_afterShareIncrease() (gas: 1298334)
5  [PASS] test_exitWithToken_badReceiver() (gas: 667244)
6  [PASS] test_exitWithToken_badTokenAddress_cappedTotalSupply() (gas: 2630849)
7  [PASS] test_exitWithToken_badTokenAddress_unlimitedTotalSupply() (gas: 2637543)
8  [PASS] test_exitWithToken_basic_asset1() (gas: 871420)
9  [PASS] test_exitWithToken_basic_asset2() (gas: 871756)
10 [PASS] test_exitWithToken_basic_debt1() (gas: 871964)
11 [PASS] test_exitWithToken_basic_debt2() (gas: 872496)
12 [PASS] test_exitWithToken_beforeShareIncrease_asset() (gas: 1298256)
13 [PASS] test_exitWithToken_differentReceiver() (gas: 887401)
14 [PASS] test_exitWithToken_fail_exitTooMuch() (gas: 664953)
15 [PASS] test_exitWithToken_fromExitWithSharesQuote() (gas: 1019869)
16 [PASS] test_exitWithToken_multiple() (gas: 1036230)
17 [PASS] test_exitWithToken_noAssetLeft() (gas: 762605)
18 [PASS] test_exitWithToken_noLiabilitiesLeft() (gas: 761811)
19 [PASS] test_exitWithToken_onBehalfOf() (gas: 1450660)
20 [PASS] test_exitWithToken_zeroAmount() (gas: 774164)
21 [PASS] test_exitWithToken_zeroSharesOwner() (gas: 75915)
22 Suite result: ok. 20 passed; 0 failed; 0 skipped; finished in 24.64ms (21.55ms CPU time)
23
24 Ran 4 tests for
   ↳ .../DynamicAssetsTokenizedBalanceSheetVault.t.sol:DynamicOrigamiTokenizedBalanceSheetVaultTestRolloverExitWithToken
25 [PASS] test_fullRollover_exitWithToken_basic_asset1New() (gas: 6069403)
26 [PASS] test_joinWithToken_multiple_exitWithAssets() (gas: 12850072)
27 [PASS] test_joinWithToken_multiple_shareIncrease_exitWithAssets() (gas: 13545622)
28 [PASS] test_joinWithToken_multiple_shareIncrease_exitWithDebt() (gas: 13538962)
29 Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 79.63ms (54.40ms CPU time)
30
31 Ran 17 tests for
   ↳ .../DynamicAssetsTokenizedBalanceSheetVault.t.sol:DynamicOrigamiTokenizedBalanceSheetVaultTestRolloverJoinWithToken
32 [PASS] test_basic_rollover() (gas: 4740073)
33 [PASS] test_joinWithToken_badTokenAddress_nonSupported_asset() (gas: 1973883)
34 [PASS] test_joinWithToken_basic_asset1New_fullRebalance() (gas: 5279272)
35 [PASS] test_joinWithToken_basic_asset1New_randomRebalance() (gas: 5505884)
36 [PASS] test_joinWithToken_basic_asset2New_fullRebalance() (gas: 5279712)
37 [PASS] test_joinWithToken_basic_asset2New_randomRebalance() (gas: 5506326)
38 [PASS] test_joinWithToken_before_full_rebalance_exit_after_emptyies_vault() (gas: 5463085)
39 [PASS] test_joinWithToken_multiple_exitWithShares() (gas: 12781891)
40 [PASS] test_joinWithToken_multiple_fullRebalance() (gas: 5461021)
41 [PASS] test_joinWithToken_multiple_rebalance_joinWithToken() (gas: 5948821)
42 [PASS] test_joinWithToken_rebalance_shareIncrease_emptyies_vault() (gas: 5604072)
43 [PASS] test_joinWithToken_singleToken_rollover() (gas: 2944903)
44 [PASS] test_maxExit() (gas: 4864403)
45 [PASS] test_previewExit_fullRebalance() (gas: 5259052)
46 [PASS] test_previewExit_halfRebalance_sharePriceDecreases() (gas: 5279997)
47 [PASS] test_previewJoin_fullRebalance() (gas: 5259130)
48 [PASS] test_previewJoin_halfRebalance_sharePrice_decreases() (gas: 5279976)
49 Suite result: ok. 17 passed; 0 failed; 0 skipped; finished in 79.75ms (54.48ms CPU time)
50
51 Ran 3 tests for
   ↳ .../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAccess
52 [PASS] test_recoverToken_access() (gas: 18072)
53 [PASS] test_seed_access() (gas: 18888)
54 [PASS] test_setMaxTotalSupply_access() (gas: 15720)
55 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 4.02ms (121.67µs CPU time)
56
57 Ran 8 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestViews
58 [PASS] test_assetTokens() (gas: 12018)
59 [PASS] test_isBalanceSheetToken_default() (gas: 21802)
60 [PASS] test_isBalanceSheetToken_updated() (gas: 1972500)
61 [PASS] test_liabilityTokens_default() (gas: 16578)
62 [PASS] test_liabilityTokens_updated() (gas: 1966817)
63 [PASS] test_supportsInterface() (gas: 19284)
64 [PASS] test_token_prices_negative() (gas: 161009)
65 [PASS] test_token_prices_positive() (gas: 136008)
66 Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 8.11ms (1.72ms CPU time)

```



```

1  Ran 1 test for
2  ↳ ../tokenizedBalanceSheet/OrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAdminNoFees
3  [PASS] test_initialization() (gas: 206301)
4  Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 3.38ms (452.33µs CPU time)
5
6  Ran 4 tests for ../olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestJoin
7  [PASS] test_join_fail_notEnough() (gas: 75909)
8  [PASS] test_join_withBorrow_existing() (gas: 583997)
9  [PASS] test_join_withBorrow_fresh() (gas: 492749)
10 [PASS] test_join_withRepay() (gas: 666608)
11 Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 6.32ms (1.60ms CPU time)
12
13 Ran 6 tests for test/foundry/unit/investments/olympus/OrigamiHOHmManager.t.sol:OrigamiHOHmManagerViews
14 [PASS] test_convertSharesToCollateral_sharesTooHigh() (gas: 14498)
15 [PASS] test_convertSharesToCollateral_someCollateralOverThreshold() (gas: 205211)
16 [PASS] test_convertSharesToCollateral_someCollateralUnderThreshold() (gas: 202932)
17 [PASS] test_convertSharesToCollateral_zeroCollateral() (gas: 14518)
18 [PASS] test_convertSharesToCollateral_zeroSupply() (gas: 14213)
19 [PASS] test_supportsInterface() (gas: 11290)
20 Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 5.93ms (503.87µs CPU time)
21
22 Ran 9 tests for ../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestExitWithShares
23 [PASS] test_exitWithShares_afterShareIncrease() (gas: 923680)
24 [PASS] test_exitWithShares_badReceiver() (gas: 658258)
25 [PASS] test_exitWithShares_basic() (gas: 844401)
26 [PASS] test_exitWithShares_beforeShareIncrease_asset() (gas: 923697)
27 [PASS] test_exitWithShares_differentReceiver() (gas: 860321)
28 [PASS] test_exitWithShares_multiple() (gas: 985226)
29 [PASS] test_exitWithShares_onBehalfOf() (gas: 1387861)
30 [PASS] test_exitWithShares_zeroAmount() (gas: 748940)
31 [PASS] test_roundTrip_checkAssetRounding() (gas: 879772)
32 Suite result: ok. 9 passed; 0 failed; 0 skipped; finished in 13.06ms (9.74ms CPU time)
33
34 Ran 8 tests for
35 ↳ ../investments/olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestMaxBorrow
36 [PASS] test_maxBorrowFromCooler_borrowDisabled() (gas: 288799)
37 [PASS] test_maxBorrowFromCooler_borrowWithNoPriorSurplus_notSavings() (gas: 471083)
38 [PASS] test_maxBorrowFromCooler_borrowWithNoPriorSurplus_savings() (gas: 466936)
39 [PASS] test_maxBorrowFromCooler_borrowWithSomePriorSurplus_savings() (gas: 477127)
40 [PASS] test_maxBorrowFromCooler_increasedOriginationLtv() (gas: 542226)
41 [PASS] test_maxBorrowFromCooler_noChange() (gas: 473169)
42 [PASS] test_maxBorrowFromCooler_noPosition() (gas: 33525)
43 [PASS] test_maxBorrowFromCooler_repay() (gas: 676082)
44 Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 7.77ms (3.15ms CPU time)
45
46 Ran 3 tests for
47 ↳ ../investments/olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestAdmin
48 [PASS] test_initialization() (gas: 138837)
49 [PASS] test_recoverToken_fail() (gas: 17866)
50 [PASS] test_recoverToken_success() (gas: 262219)
51 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 7.85ms (535.29µs CPU time)
52
53 Ran 5 tests for
54 ↳ ../olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestSavings
55 [PASS] test_syncDebtTokenSavings_deposit() (gas: 256618)
56 [PASS] test_syncDebtTokenSavings_noDifference() (gas: 386085)
57 [PASS] test_syncDebtTokenSavings_savingsNotSet() (gas: 4151080)
58 [PASS] test_syncDebtTokenSavings_withdraw_noCap() (gas: 409393)
59 [PASS] test_syncDebtTokenSavings_withdraw_zeroMaxWithdraw() (gas: 383589)
60 Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 9.21ms (2.72ms CPU time)
61
62 Ran 3 tests for
63 ↳ ../investments/olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestExit
64 [PASS] test_exit_withBorrow() (gas: 637756)
65 [PASS] test_exit_withRepay_existing() (gas: 701399)
66 [PASS] test_exit_withRepay_fresh() (gas: 606225)
67 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 7.18ms (2.47ms CPU time)
68
69 Ran 3 tests for ../olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestSweep
70 [PASS] test_sweep_fail_tooMuch() (gas: 38585)
71 [PASS] test_sweep_noSavingsVault() (gas: 464668)
72 [PASS] test_sweep_withSavingsVault() (gas: 429666)
73 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 5.97ms (1.15ms CPU time)

```

```

1 Ran 1 test for .../olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestExitAfterChange
2 [PASS] test_exit_afterChange() (gas: 2651785)
3 Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 6.46ms (1.19ms CPU time)
4
5 Ran 14 tests for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestExitWithToken
6 [PASS] test_exitWithShares_fromExitWithTokenQuote() (gas: 844714)
7 [PASS] test_exitWithShares_fullUnwind() (gas: 1030198)
8 [PASS] test_exitWithToken_afterShareIncrease() (gas: 944778)
9 [PASS] test_exitWithToken_badReceiver() (gas: 661060)
10 [PASS] test_exitWithToken_basic_asset1() (gas: 864748)
11 [PASS] test_exitWithToken_basic_asset2() (gas: 865101)
12 [PASS] test_exitWithToken_basic_debt1() (gas: 865428)
13 [PASS] test_exitWithToken_basic_debt2() (gas: 866019)
14 [PASS] test_exitWithToken_beforeShareIncrease_asset() (gas: 944511)
15 [PASS] test_exitWithToken_differentReceiver() (gas: 880701)
16 [PASS] test_exitWithToken_fromExitWithSharesQuote() (gas: 876037)
17 [PASS] test_exitWithToken_multiple() (gas: 1027274)
18 [PASS] test_exitWithToken_onBehalfOf() (gas: 1437784)
19 [PASS] test_exitWithToken_zeroAmount() (gas: 769224)
20 Suite result: ok. 14 passed; 0 failed; 0 skipped; finished in 21.78ms (18.37ms CPU time)
21
22 Ran 1 test for .../olympus/OrigamiHOHmManagerDebtTokenChange.t.sol:OrigamiHOHmManagerDebtTokenChangeTestViews
23 [PASS] test_convertSharesToCollateral_someCollateral() (gas: 205175)
24 Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 6.20ms (136.88µs CPU time)
25
26 Ran 1 test for
27   ↳ .../tokenizedBalanceSheet/VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestPermit
28 [PASS] test_permit() (gas: 90962)
29 Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 5.01ms (440.25µs CPU time)
30
31 Ran 2 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestAccess
32 [PASS] test_setManager_access() (gas: 17852)
33 [PASS] test_setTokenPrices_access() (gas: 17842)
34 Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 5.02ms (90.96µs CPU time)
35
36 Ran 5 tests for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestViewsNoFees
37 [PASS] test_maxExit() (gas: 143358)
38 [PASS] test_maxJoin() (gas: 83516)
39 [PASS] test_noFees() (gas: 15253121)
40 [PASS] test_previewExit() (gas: 160772)
41 [PASS] test_previewJoin() (gas: 160725)
42 Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 7.46ms (4.20ms CPU time)
43
44 Ran 10 tests for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestJoinWithShares
45 [PASS] test_joinWithShares_afterShareIncrease() (gas: 734368)
46 [PASS] test_joinWithShares_badReceiver() (gas: 522123)
47 [PASS] test_joinWithShares_basic() (gas: 643275)
48 [PASS] test_joinWithShares_beforeShareIncrease_asset() (gas: 721722)
49 [PASS] test_joinWithShares_differentReceiver() (gas: 648137)
50 [PASS] test_joinWithShares_exactlyOneShare() (gas: 643219)
51 [PASS] test_joinWithShares_multiple() (gas: 944780)
52 [PASS] test_joinWithShares_partialShare() (gas: 658237)
53 [PASS] test_joinWithShares_unlimited_shares_capacity() (gas: 603694)
54 [PASS] test_joinWithShares_zeroAmount() (gas: 511410)
55 Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 12.66ms (8.86ms CPU time)
56
57 Ran 8 tests for test/foundry/unit/investments/olympus/OrigamiHOHmVault.t.sol:OrigamiHOHmVaultTestAdmin
58 [PASS] test_initialization() (gas: 378214)
59 [PASS] test_seed() (gas: 9703858)
60 [PASS] test_setManager_fail() (gas: 13715)
61 [PASS] test_setManager_newDebtToken() (gas: 1715699)
62 [PASS] test_setManager_newManager() (gas: 4077134)
63 [PASS] test_setManager_sameManager() (gas: 29718)
64 [PASS] test_setTokenPrices_fail() (gas: 13641)
65 [PASS] test_setTokenPrices_success() (gas: 24394)
Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 8.18ms (2.70ms CPU time)

```

```

1  Ran 13 tests for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestJoinWithToken
2  [PASS] test_joinWithToken_afterShareIncrease() (gas: 741029)
3  [PASS] test_joinWithToken_badReceiver() (gas: 524047)
4  [PASS] test_joinWithToken_badTokenAddress_cappedTotalSupply() (gas: 2168913)
5  [PASS] test_joinWithToken_badTokenAddress_unlimitedTotalSupply() (gas: 2168914)
6  [PASS] test_joinWithToken_basic_asset1() (gas: 662492)
7  [PASS] test_joinWithToken_basic_asset2() (gas: 662863)
8  [PASS] test_joinWithToken_basic_debt1() (gas: 663582)
9  [PASS] test_joinWithToken_basic_debt2() (gas: 663918)
10 [PASS] test_joinWithToken_beforeShareIncrease_asset() (gas: 764680)
11 [PASS] test_joinWithToken_differentReceiver() (gas: 667387)
12 [PASS] test_joinWithToken_multiple() (gas: 985045)
13 [PASS] test_joinWithToken_overflows_on_weird_tokens_that_perform_maxBalance_transfer_on_uint256Max() (gas: 420410)
14 [PASS] test_joinWithToken_zeroAmount() (gas: 530609)
15 Suite result: ok. 13 passed; 0 failed; 0 skipped; finished in 13.10ms (9.27ms CPU time)
16
17 Ran 1 test for
18   ↳ .../tokenizedBalanceSheet/VanillaOrigamiTokenizedBalanceSheetVault.t.sol:CustomOrigamiTokenizedBalanceSheetVaultTest
19 [PASS] test_roundSharesTo0() (gas: 33988316)
20 Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 60.75ms (57.50ms CPU time)
21
22 Ran 3 tests for
23   ↳ .../tokenizedBalanceSheet/VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAccess
24 [PASS] test_recoverToken_access() (gas: 18072)
25 [PASS] test_seed_access() (gas: 18888)
26 [PASS] test_setMaxTotalSupply_access() (gas: 15720)
27 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 3.34ms (123.46µs CPU time)
28
29 Ran 1 test for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAdminNoFees
30 [PASS] test_initialization() (gas: 206303)
31 Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 3.79ms (448.92µs CPU time)
32
33 Ran 9 tests for .../VanillaOrigamiTokenizedBalanceSheetVault.t.sol:OrigamiTokenizedBalanceSheetVaultTestAdminWithFees
34 [PASS] test_initialization() (gas: 223033)
35 [PASS] test_recoverToken_assetsAndLiabilities() (gas: 777589)
36 [PASS] test_recoverToken_success() (gas: 2039009)
37 [PASS] test_seedDeposit_failure_badParams() (gas: 14148881)
38 [PASS] test_seedDeposit_failure_maxTooLow() (gas: 14142752)
39 [PASS] test_seedDeposit_success() (gas: 14698440)
40 [PASS] test_seed_failure_alreadySeeded() (gas: 18673)
41 [PASS] test_setMaxTotalSupply_failure_zeroSupply() (gas: 4461465)
42 [PASS] test_setMaxTotalSupply_success() (gas: 23664)
43 Suite result: ok. 9 passed; 0 failed; 0 skipped; finished in 10.82ms (7.12ms CPU time)
44
45 Ran 7 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerExit
46 [PASS] test_exit_fail_tooMuchUndelegationAmount() (gas: 937159)
47 [PASS] test_exit_success_paused() (gas: 643037)
48 [PASS] test_exit_withBorrow() (gas: 675424)
49 [PASS] test_exit_withDelegate_overThreshold() (gas: 992645)
50 [PASS] test_exit_withDelegate_underThreshold() (gas: 1130614)
51 [PASS] test_exit_withRepay_existing() (gas: 789936)
52 [PASS] test_exit_withRepay_fresh() (gas: 652057)
53 Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 8.97ms (4.71ms CPU time)
54
55 Ran 7 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerJoin
56 [PASS] test_join_fail_notEnough() (gas: 75905)
57 [PASS] test_join_success_paused() (gas: 507353)
58 [PASS] test_join_withBorrow_existing() (gas: 641774)
59 [PASS] test_join_withBorrow_fresh() (gas: 509834)
60 [PASS] test_join_withDelegate_overThreshold() (gas: 829201)
61 [PASS] test_join_withDelegate_underThreshold() (gas: 666792)
62 [PASS] test_join_withRepay() (gas: 701214)
63 Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 7.17ms (2.94ms CPU time)
64
65 Ran 15 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerTestDelegations
66 [PASS] test_setDelegationAmount1_increase_decrease() (gas: 570536)
67 [PASS] test_setDelegationAmount1_noChange() (gas: 553893)
68 [PASS] test_setDelegationAmount1_noCollateral() (gas: 37386)
69 [PASS] test_setDelegationAmount1_noPrior() (gas: 229414)
70 [PASS] test_setDelegationAmount2_increase_decrease() (gas: 570076)
71 [PASS] test_setDelegationAmount2_noChange() (gas: 909102)
72 [PASS] test_setDelegationAmount2_noPrior() (gas: 248764)
73 [PASS] test_updateDelegateAndAmount_manyUsers() (gas: 317247137)

```

```

1  [PASS] test_updateDelegateAndAmount_noCollateral() (gas: 60592)
2  [PASS] test_updateDelegateAndAmount_ohmBackingIncrease() (gas: 632409)
3  [PASS] test_updateDelegateAndAmount_other() (gas: 778059)
4  [PASS] test_updateDelegateAndAmount_remove() (gas: 539345)
5  [PASS] test_updateDelegateAndAmount_self() (gas: 557786)
6  [PASS] test_updateDelegateAndAmount_zeroShares() (gas: 244613)
7  [PASS] test_updateDelegateAndAmount_zeroSupply() (gas: 244285)
8  Suite result: ok. 15 passed; 0 failed; 0 skipped; finished in 205.52ms (179.40ms CPU time)
9
10 Ran 8 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerTestSavings
11 [PASS] test_syncDebtTokenSavings_depositNoCap() (gas: 256622)
12 [PASS] test_syncDebtTokenSavings_deposit_smallMaxDeposit() (gas: 255470)
13 [PASS] test_syncDebtTokenSavings_deposit_zeroMaxDeposit() (gas: 208109)
14 [PASS] test_syncDebtTokenSavings_noDifference() (gas: 386155)
15 [PASS] test_syncDebtTokenSavings_savingsNotSet() (gas: 4150917)
16 [PASS] test_syncDebtTokenSavings_withdraw_noCap() (gas: 409418)
17 [PASS] test_syncDebtTokenSavings_withdraw_smallMaxWithdraw() (gas: 406877)
18 [PASS] test_syncDebtTokenSavings_withdraw_zeroMaxWithdraw() (gas: 383659)
19 Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 12.70ms (5.50ms CPU time)
20
21 Ran 6 tests for test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerTestSweep
22 [PASS] test_sweepCallback_noBalance() (gas: 30963)
23 [PASS] test_sweepCallback_smallBalance() (gas: 430316)
24 [PASS] test_sweep_fail_cooldown() (gas: 525580)
25 [PASS] test_sweep_fail_tooMuch() (gas: 38607)
26 [PASS] test_sweep_noSavingsVault() (gas: 480390)
27 [PASS] test_sweep_withSavingsVault() (gas: 429684)
28 Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 7.24ms (2.23ms CPU time)
29
30 Ran 5 tests for ...:OrigamiTokenizedBalanceSheetVaultTestJoinExitSharesRounding
31 [PASS] test_previewExit1() (gas: 15422813)
32 [PASS] test_previewJoinExit1() (gas: 152621196)
33 [PASS] test_previewJoinExit2() (gas: 151943653)
34 [PASS] test_previewJoinExit3() (gas: 30686663)
35 [PASS] test_previewJoinExit_fuzz(uint256,uint96,uint96,uint96,uint96,uint96,uint96) (runs: 512, : 15476214, ~: 15477349)
36 Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 1.87s (1.86s CPU time)
37
38 Ran 3 tests for test/foundry/unit/investments/olympus/OrigamiCoolerMigrator.t.sol:OrigamiCoolerMigratorTestAccess
39 [PASS] test_access_onFlashLoan_invalid_caller() (gas: 14052)
40 [PASS] test_access_onFlashLoan_invalid_initiator() (gas: 16273)
41 [PASS] test_access_setMaxLoans() (gas: 15676)
42 Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 3.61s (460.71µs CPU time)
43
44 Ran 2 tests for test/foundry/unit/investments/olympus/OrigamiCoolerMigrator.t.sol:OrigamiCoolerMigratorTestAdmin
45 [PASS] test_init() (gas: 49217)
46 [PASS] test_settMaxLoans() (gas: 20337)
47 Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 3.54s (556.71µs CPU time)
48
49 Ran 7 tests for
50   ↳ test/foundry/unit/investments/olympus/OrigamiHOhmManager.t.sol:OrigamiHOhmManagerTestDelegationThreshold
51 [PASS] test_setDelegationAmount1_underThreshold_fromNoDelegation() (gas: 259154)
52 [PASS] test_setDelegationAmount1_underThreshold_fromSomeDelegation() (gas: 536909)
53 [PASS] test_setDelegationAmount2_underThreshold_fromNoDelegation() (gas: 919253)
54 [PASS] test_setDelegationAmount2_underThreshold_fromSomeDelegation() (gas: 832291)
55 [PASS] test_updateDelegateAndAmount_atThreshold() (gas: 557701)
56 [PASS] test_updateDelegateAndAmount_underThreshold_fromNoDelegation() (gas: 244639)
57 [PASS] test_updateDelegateAndAmount_underThreshold_fromSomeDelegation() (gas: 537479)
58 Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 3.54s (2.26ms CPU time)
59
60 Ran 11 tests for test/foundry/unit/investments/olympus/OrigamiCoolerMigrator.t.sol:OrigamiCoolerMigratorTestView
61 [PASS] test_getCoolerLoansFor_alreadyPaidDown() (gas: 705507)
62 [PASS] test_getCoolerLoansFor_expired() (gas: 705918)
63 [PASS] test_getCoolerLoansFor_invalidLoanLender() (gas: 707523)
64 [PASS] test_getCoolerLoansFor_only_monocooler() (gas: 384521)
65 [PASS] test_getCoolerLoansFor_unhandledPanic() (gas: 483562)
66 [PASS] test_getCoolerLoansFor_v1_and_mono_coolers() (gas: 1079647)
67 [PASS] test_getCoolerLoansFor_v1_coolers() (gas: 607156)
68 [PASS] test_getCoolerLoansFor_v1_coolers_failInvalidCooler() (gas: 34060)
69 [PASS] test_getCoolerV1_1Params() (gas: 19881)
70 [PASS] test_previewMigration_all_coolers() (gas: 988710)
71 [PASS] test_previewMigration_v1_cooler_multiple_loans() (gas: 327353)
72 Suite result: ok. 11 passed; 0 failed; 0 skipped; finished in 3.71s (99.28ms CPU time)

```

```

1 Ran 29 tests for test/foundry/unit/investments/olympus/OrigamiCoolerMigrator.t.sol:OrigamiCoolerMigratorTest
2 [PASS] test_migrate_coolers_all() (gas: 2368717)
3 [PASS] test_migrate_coolers_fail_slippage_shares_surplus() (gas: 1111325)
4 [PASS] test_migrate_coolers_fail_slippage_usds_shortfall() (gas: 852721)
5 [PASS] test_migrate_coolers_fail_slippage_usds_surplus() (gas: 1111417)
6 [PASS] test_migrate_coolers_filtered() (gas: 1777052)
7 [PASS] test_migrate_coolers_monocooler_fail_delegations() (gas: 2535739)
8 [PASS] test_migrate_coolers_monocooler_success_delegations_and_removed() (gas: 2995958)
9 [PASS] test_migrate_coolers_success_slippage_shortfall() (gas: 1765212)
10 [PASS] test_migrate_coolers_success_slippage_surplus() (gas: 2338307)
11 [PASS] test_migrate_fail_coolerv1_expired() (gas: 629326)
12 [PASS] test_migrate_fail_coolerv1_fullyRepaid() (gas: 628947)
13 [PASS] test_migrate_fail_coolerv1_notExpectedLender() (gas: 631062)
14 [PASS] test_migrate_fail_coolerv1_otherRevert() (gas: 623772)
15 [PASS] test_migrate_fail_coolerv1_outOfBounds() (gas: 621832)
16 [PASS] test_migrate_fail_coolerv1_unhandledPanic() (gas: 622585)
17 [PASS] test_migrate_fail_monocooler_invalidOwner() (gas: 410768)
18 [PASS] test_migrate_fail_monocooler_misMatchedCollateralWithdrawal() (gas: 965300)
19 [PASS] test_migrate_fail_monocooler_misMatchedDebtRepayment() (gas: 809504)
20 [PASS] test_migrate_fail_monocooler_paused() (gas: 18540)
21 [PASS] test_migrate_fail_monocooler_wrongAuthorized() (gas: 418404)
22 [PASS] test_migrate_fail_monocooler_wrongCaller() (gas: 418226)
23 [PASS] test_migrate_fail_monocooler_zeroDebt() (gas: 1619185)
24 [PASS] test_migrate_fail_nothingToMigrate() (gas: 109369)
25 [PASS] test_migrate_monocooler_exact() (gas: 1463689)
26 [PASS] test_migrate_monocooler_receiveSurplus() (gas: 1457633)
27 [PASS] test_migrate_monocooler_shortfall() (gas: 1765313)
28 [PASS] test_migrate_v1_coolers_fail_notCoolerOwner() (gas: 419040)
29 [PASS] test_migrate_v1_coolers_fail_notFromFactory() (gas: 30594)
30 [PASS] test_migrate_v1_coolers_noPositions() (gas: 97264)
31 Suite result: ok. 29 passed; 0 failed; 0 skipped; finished in 4.01s (401.52ms CPU time)
32
33 Ran 28 tests for
34   ↳ .../DynamicAssetsTokenizedBalanceSheetVaultFuzzTest.t.sol:DynamicAssetsTokenizedBalanceSheetVaultFuzzTest
35 [PASS] test_RT_exitWithShares_joinWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 9304464, ~:
36   ↳ 9309748)
37 [PASS] test_RT_exitWithShares_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
38   ↳ 9007855, ~: 9018349)
39 [PASS] test_RT_exitWithToken_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
40   ↳ 9318304, ~: 9342886)
41 [PASS] test_RT_joinWithShares_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
42   ↳ 9323527, ~: 9339447)
43 [PASS] test_RT_joinWithShares_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
44   ↳ 9344344, ~: 9350638)
45 [PASS] test_RT_joinWithToken_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
46   ↳ 9309959, ~: 9324351)
47 [PASS] test_RT_joinWithToken_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
48   ↳ 9333747, ~: 9347428)
49 [PASS] test_RT_previewExitWithToken_exactOutput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
50   ↳ 8553514, ~: 8559512)
51 [PASS] test_RT_previewJoinWithToken_exactInput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
52   ↳ 8551777, ~: 8559265)
53 [PASS] test_RT_round_trips_tokens((address[4],uint256[4],uint32[4]),bool,uint32,uint256,uint32) (runs: 512, : 8643324,
54   ↳ ~: 8648983)
55 [PASS] test_RT_withdraw_deposit((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 9320629, ~:
56   ↳ 9345164)
57 [PASS] test_balanceSheet((address[4],uint256[4],uint32[4])) (runs: 512, : 55084, ~: 55084)
58 [PASS] test_convertFromShares((address[4],uint256[4],uint32[4]),uint128) (runs: 512, : 8553742, ~: 8558008)
59 [PASS] test_convertFromToken((address[4],uint256[4],uint32[4]),bool,uint32,uint128) (runs: 512, : 8570968, ~: 8575291)
60 [PASS] test_exitWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 8873274, ~: 8879563)
61 [PASS] test_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 8882743, ~: 8896562)
62 [PASS] test_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 9053073, ~: 9058274)
63 [PASS] test_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 8713508, ~: 8719042)
64 [PASS] test_maxExitWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 8522220, ~: 8525742)
65 [PASS] test_maxExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 72741, ~: 72540)
66 [PASS] test_maxJoinWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 8523009, ~: 8526410)
67 [PASS] test_maxJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 8537079, ~: 8542395)
68 [PASS] test_previewExitWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 8870153, ~: 8875483)
69 [PASS] test_previewExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 8892439, ~:
70   ↳ 8906510)

```



```

1  [PASS] test_previewJoinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 9044547, ~:
   ↳ 9050216)
2  [PASS] test_previewJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 8707852, ~:
   ↳ 8713287)
3  [PASS] test_tokens((address[4],uint256[4],uint32[4])) (runs: 512, : 8524874, ~: 8530048)
4  [PASS] test_unsuccessful_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 8630495, ~:
   ↳ 8630609)
5  Suite result: ok. 28 passed; 0 failed; 0 skipped; finished in 38.40s (128.29s CPU time)
6
7  Ran 32 tests for
   ↳ .../VanillaOrigamiTokenizedBalanceSheetVaultFuzzTest.t.sol:VanillaOrigamiTokenizedBalanceSheetVaultFuzzTest
8  [PASS] invariant_previewExitWithTokenAlwaysGiveExactOutput() (runs: 128, calls: 10240, reverts: 0)
9  [PASS] invariant_previewJoinWithTokenAlwaysTakesExactInput() (runs: 128, calls: 10240, reverts: 0)
10 [PASS] invariant_totalAssetsConsistency() (runs: 128, calls: 10240, reverts: 0)
11 [PASS] invariant_whenYieldIncreasesAssetSharePriceAlsoDoes() (runs: 128, calls: 10240, reverts: 0)
12 [PASS] test_RT_exitWithShares_joinWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 3896561, ~:
   ↳ 3902313)
13 [PASS] test_RT_exitWithShares_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3736982, ~: 3740544)
14 [PASS] test_RT_exitWithToken_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3899749, ~: 3918942)
15 [PASS] test_RT_exitWithShares_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3873502, ~: 3853758)
16 [PASS] test_RT_joinWithShares_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3888261, ~: 3864632)
17 [PASS] test_RT_joinWithToken_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3859166, ~: 3842177)
18 [PASS] test_RT_joinWithToken_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3878130, ~: 3861795)
19 [PASS] test_RT_previewExitWithToken_exactOutput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3371743, ~: 3375045)
20 [PASS] test_RT_previewJoinWithToken_exactInput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3370623, ~: 3374940)
21 [PASS] test_RT_round_trips_tokens((address[4],uint256[4],uint32[4]),bool,uint32,uint256,uint32) (runs: 512, : 3422189,
   ↳ ~: 3426118)
22 [PASS] test_RT_withdraw_deposit((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3896539, ~:
   ↳ 3920684)
23 [PASS] test_balanceSheet((address[4],uint256[4],uint32[4])) (runs: 512, : 55106, ~: 55106)
24 [PASS] test_convertFromShares((address[4],uint256[4],uint32[4]),uint128) (runs: 512, : 3378797, ~: 3381334)
25 [PASS] test_convertFromToken((address[4],uint256[4],uint32[4]),bool,uint32,uint128) (runs: 512, : 3383844, ~: 3386680)
26 [PASS] test_exitWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 3706605, ~: 3708207)
27 [PASS] test_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3702750, ~: 3716079)
28 [PASS] test_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3618929, ~: 3611234)
29 [PASS] test_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3431986, ~: 3436299)
30 [PASS] test_maxExitWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 3354730, ~: 3358113)
31 [PASS] test_maxExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 45052, ~: 43817)
32 [PASS] test_maxJoinWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 3355232, ~: 3358245)
33 [PASS] test_maxJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 3358492, ~: 3362213)
34 [PASS] test_previewExitWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 3704257, ~: 3705747)
35 [PASS] test_previewExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3712442, ~:
   ↳ 3721854)
36 [PASS] test_previewJoinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3614037, ~:
   ↳ 3605463)
37 [PASS] test_previewJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3427367, ~:
   ↳ 3432070)
38 [PASS] test_tokens((address[4],uint256[4],uint32[4])) (runs: 512, : 3357435, ~: 3361089)
39 [PASS] test_unsuccessful_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3441336, ~:
   ↳ 3441963)
40 Suite result: ok. 32 passed; 0 failed; 0 skipped; finished in 40.23s (148.47s CPU time)
41
42 Ran 33 tests for .../VanillaOrigamiTokenizedBalanceSheetVaultFuzzTestWithoutSharePrice
43 [PASS] invariant_alwaysHoldsTheExchangeRateUntilYield() (runs: 128, calls: 10240, reverts: 0)
44 [PASS] invariant_previewExitWithTokenAlwaysGiveExactOutput() (runs: 128, calls: 10240, reverts: 0)
45 [PASS] invariant_previewJoinWithTokenAlwaysTakesExactInput() (runs: 128, calls: 10240, reverts: 0)
46 [PASS] invariant_totalAssetsConsistency() (runs: 128, calls: 10240, reverts: 0)
47 [PASS] invariant_whenYieldIncreasesAssetSharePriceAlsoDoes() (runs: 128, calls: 10240, reverts: 0)
48 [PASS] test_RT_exitWithShares_joinWithShares((address[4],uint256[4],uint32[4]),uint256) (runs: 512, : 3896307, ~:
   ↳ 3902029)
49 [PASS] test_RT_exitWithShares_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3738010, ~: 3740550)
50 [PASS] test_RT_exitWithToken_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3899338, ~: 3918982)

```

```

1  [PASS] test_RT_joinWithShares_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3871890, ~: 3853320)
2  [PASS] test_RT_joinWithShares_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3886885, ~: 3864621)
3  [PASS] test_RT_joinWithToken_exitWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3861891, ~: 3842689)
4  [PASS] test_RT_joinWithToken_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3881208, ~: 3862017)
5  [PASS] test_RT_previewExitWithToken_exactOutput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3371839, ~: 3374992)
6  [PASS] test_RT_previewJoinWithToken_exactInput((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, :
   ↳ 3371136, ~: 3375106)
7  [PASS] test_RT_withdraw_deposit((address[4],uint256[4],uint32[4]),bool,uint32,uint256,uint32) (runs: 512, : 3422937,
   ↳ ~: 3425928)
8  [PASS] test_RT_withdraw_deposit((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3899573, ~:
   ↳ 3921172)
9  [PASS] test_balanceSheet((address[4],uint256[4],uint32[4])) (runs: 512, : 55084, ~: 55084)
10 [PASS] test_convertFromShares((address[4],uint256[4],uint32[4],uint128) (runs: 512, : 3378819, ~: 3381443)
11 [PASS] test_convertFromToken((address[4],uint256[4],uint32[4]),bool,uint32,uint128) (runs: 512, : 3383408, ~: 3386890)
12 [PASS] test_exitWithShares((address[4],uint256[4],uint32[4],uint256) (runs: 512, : 3707775, ~: 3708583)
13 [PASS] test_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3705422, ~: 3716165)
14 [PASS] test_joinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3617358, ~: 3610882)
15 [PASS] test_joinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3431540, ~: 3436430)
16 [PASS] test_maxExitWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 3354388, ~: 3358252)
17 [PASS] test_maxExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 45074, ~: 43842)
18 [PASS] test_maxJoinWithShares((address[4],uint256[4],uint32[4])) (runs: 512, : 3355272, ~: 3358063)
19 [PASS] test_maxJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32) (runs: 512, : 3359327, ~: 3362413)
20 [PASS] test_previewExitWithShares((address[4],uint256[4],uint32[4],uint256) (runs: 512, : 3703678, ~: 3705743)
21 [PASS] test_previewExitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3711096, ~:
   ↳ 3722055)
22 [PASS] test_previewJoinWithShares((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3613441, ~:
   ↳ 3605391)
23 [PASS] test_previewJoinWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3428877, ~:
   ↳ 3432208)
24 [PASS] test_tokens((address[4],uint256[4],uint32[4])) (runs: 512, : 3356294, ~: 3361078)
25 [PASS] test_unsuccessful_exitWithToken((address[4],uint256[4],uint32[4]),bool,uint32,uint256) (runs: 512, : 3440603, ~:
   ↳ 3441731)
26 Suite result: ok. 33 passed; 0 failed; 0 skipped; finished in 40.14s (160.02s CPU time)
27
28 Ran 57 test suites in 44.21s (140.04s CPU time): 512 tests passed, 0 failed, 0 skipped (512 total tests)

```

8.3 Automated Tools

8.3.1 AuditAgent

All the relevant issues raised by the AuditAgent have been incorporated into this report. The AuditAgent is an AI-powered smart contract auditing tool that analyses code, detects vulnerabilities, and provides actionable fixes. It accelerates the security analysis process, complementing human expertise with advanced AI models to deliver efficient and comprehensive smart contract audits. Available at <https://app.auditagent.nethermind.io>.

9 About Nethermind

Nethermind is a Blockchain Research and Software Engineering company. Our work touches every part of the web3 ecosystem - from layer 1 and layer 2 engineering, cryptography research, and security to application-layer protocol development. We offer strategic support to our institutional and enterprise partners across the blockchain, digital assets, and DeFi sectors, guiding them through all stages of the research and development process, from initial concepts to successful implementation.

We offer security audits of projects built on EVM-compatible chains and Starknet. We are active builders of the Starknet ecosystem, delivering a node implementation, a block explorer, a Solidity-to-Cairo transpiler, and formal verification tooling. Nethermind also provides strategic support to our institutional and enterprise partners in blockchain, digital assets, and decentralized finance (DeFi). In the next paragraphs, we introduce the company in more detail.

Blockchain Security: At Nethermind, we believe security is vital to the health and longevity of the entire Web3 ecosystem. We provide security services related to Smart Contract Audits, Formal Verification, and Real-Time Monitoring. Our Security Team comprises blockchain security experts in each field, often collaborating to produce comprehensive and robust security solutions. The team has a strong academic background, can apply state-of-the-art techniques, and is experienced in analyzing cutting-edge Solidity and Cairo smart contracts, such as ArgentX and StarkGate (the bridge connecting Ethereum and StarkNet). Most team members hold a Ph.D. degree and actively participate in the research community, accounting for 240+ articles published and 1,450+ citations in Google Scholar. The security team adopts customer-oriented and interactive processes where clients are involved in all stages of the work.

Blockchain Core Development: Our core engineering team, consisting of over 20 developers, maintains, improves, and upgrades our flagship product - the Nethermind Ethereum Execution Client. The client has been successfully operating for several years, supporting both the Ethereum Mainnet and its testnets, and now accounts for nearly a quarter of all synced Mainnet nodes. Our unwavering commitment to Ethereum's growth and stability extends to sidechains and layer 2 solutions. Notably, we were the sole execution layer client to facilitate Gnosis Chain's Merge, transitioning from Aura to Proof of Stake (PoS), and we are actively developing a full-node client to bolster Starknet's decentralization efforts. Our core team equips partners with tools for seamless node set-up, using generated docker-compose scripts tailored to their chosen execution client and preferred configurations for various network types.

DevOps and Infrastructure Management: Our infrastructure team ensures our partners' systems operate securely, reliably, and efficiently. We provide infrastructure design, deployment, monitoring, maintenance, and troubleshooting support, allowing you to focus on your core business operations. Boasting extensive expertise in Blockchain as a Service, private blockchain implementations, and node management, our infrastructure and DevOps engineers are proficient with major cloud solution providers and can host applications in-house or on clients' premises. Our global in-house SRE teams offer 24/7 monitoring and alerts for both infrastructure and application levels. We manage over 5,000 public and private validators and maintain nodes on major public blockchains such as Polygon, Gnosis, Solana, Cosmos, Near, Avalanche, Polkadot, Aptos, and StarkWare L2. Sedge is an open-source tool developed by our infrastructure experts, designed to simplify the complex process of setting up a proof-of-stake (PoS) network or chain validator. Sedge generates docker-compose scripts for the entire validator set-up based on the chosen client, making the process easier and quicker while following best practices to avoid downtime and being slashed.

Cryptography Research: At Nethermind, our Cryptography Research team is dedicated to continuous internal research while fostering close collaboration with external partners. The team has expertise across a wide range of domains, including cryptography protocols, consensus design, decentralized identity, verifiable credentials, Sybil resistance, oracles, and credentials, distributed validator technology (DVT), and Zero-knowledge proofs. This diverse skill set, combined with strong collaboration between our engineering teams, enables us to deliver cutting-edge solutions to our partners and clients.

Smart Contract Development & DeFi Research: Our smart contract development and DeFi research team comprises 40+ world-class engineers who collaborate closely with partners to identify needs and work on value-adding projects. The team specializes in Solidity and Cairo development, architecture design, and DeFi solutions, including DEXs, AMMs, structured products, derivatives, and money market protocols, as well as ERC20, 721, and 1155 token design. Our research and data analytics focuses on three key areas: technical due diligence, market research, and DeFi research. Utilizing a data-driven approach, we offer in-depth insights and outlooks on various industry themes.

Our suite of L2 tooling: Warp is Starknet's approach to EVM compatibility. It allows developers to take their Solidity smart contracts and transpile them to Cairo, Starknet's smart contract language. In the short time since its inception, the project has accomplished many achievements, including successfully transpiling Uniswap v3 onto Starknet using Warp.

- **Voyager** is a user-friendly Starknet block explorer that offers comprehensive insights into the Starknet network. With its intuitive interface and powerful features, Voyager allows users to easily search for and examine transactions, addresses, and contract details. As an essential tool for navigating the Starknet ecosystem, Voyager is the go-to solution for users seeking in-depth information and analysis;
- **Horus** is an open-source formal verification tool for StarkNet smart contracts. It simplifies the process of formally verifying Starknet smart contracts, allowing developers to express various assertions about the behavior of their code using a simple assertion language;
- **Juno** is a full-node client implementation for Starknet, drawing on the expertise gained from developing the Nethermind Client. Written in Golang and open-sourced from the outset, Juno verifies the validity of the data received from Starknet by comparing it to proofs retrieved from Ethereum, thus maintaining the integrity and security of the entire ecosystem.

Learn more about us at nethermind.io.

General Advisory to Clients

As auditors, we recommend that any changes or updates made to the audited codebase undergo a re-audit or security review to address potential vulnerabilities or risks introduced by the modifications. By conducting a re-audit or security review of the modified codebase, you can significantly enhance the overall security of your system and reduce the likelihood of exploitation. However, we do not possess the authority or right to impose obligations or restrictions on our clients regarding codebase updates, modifications, or subsequent audits. Accordingly, the decision to seek a re-audit or security review lies solely with you.

Disclaimer

This report is based on the scope of materials and documentation provided by you to [Nethermind](#) in order that [Nethermind](#) could conduct the security review outlined in **1. Executive Summary** and **2. Audited Files**. The results set out in this report may not be complete nor inclusive of all vulnerabilities. [Nethermind](#) has provided the review and this report on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. This report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on this report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, [Nethermind](#) disclaims any liability in connection with this report, its content, and any related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. [Nethermind](#) does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and [Nethermind](#) will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.