# Thruster AMM Audits Change Log

## Blast Changes to Contracts

There are two Blast specific contracts added, `ThrusterGas.sol` and `ThrusterYield.sol`.

- The exception is that for V3 on the `ThrusterPoolDeployer.sol` and `ThrusterPool.sol` contracts, they do not inherit these Blast contracts and instead directly configure the Blast specific configurations due to contract size limitations.

There are fields marked as `Yield` and `Gas` in each table. This will indicate if the contract inherits the `ThrusterYield.sol` contract or `ThrusterGas.sol`, respectively. `ThrusterYield.sol` handles both rebasing yield and gas claiming, while `ThrusterGas.sol` only handles gas claiming.

## CFMM - Uniswap V2 Changes

- `PAIR_INIT_HASH_CODE` = `0xccf872df471830af5eb8a3a0293a92146c157aa68d980a34add2d26663a4caff`
- Unless explicitly mentioned, contract names and abis are consistent with Uniswap V2
- Changed all contracts and interfaces with `UniswapV2` to `Thruster`
- Modifiable protocol fee if turned on between 16% ~ 50% of the LP fee (always 0.3%), so ~0.05% ~ 0.15% protocol fees earned per swap. If off, then there is no protocol fee, i.e. 0%.
- Modified `feeToSetter` to `owner` to also control claimable yield on Blast

**V2 Changelogs**

| Aa Name | Fork | Yield | Changes |
|---|---|---|---|
| ThrusterFactory.sol | UniswapV2Factory.sol | True | 1. Modified name “fee” to “yield”<br>2. Added dynamic protocol yield cut `yieldCut` that is modifiable by `yieldToSetter`<br>3. Added events for state changes<br>4. Changed names from “UniswapV2” to “Thruster”<br>5. Added `setYieldCut` method, values between 1 to 6 to set yield fee amount. To remove fee, set the YieldToSetterAddress as 0<br>6. Emitting events from the Pair contract on the Factory contract to make |
| ThrusterPair.sol | UniswapV2Pair.sol | True | 1. Modified names from “UniswapV2” to “Thruster”<br>2. Renamed `_mintFee` to `_mintYieldCut`<br>3. Removed hardcoded `1/6` fee value to directly fetch fee portion from the factory by calling `yieldCut`<br>4. Renamed `feeOn` to `yieldCutOn`<br>5. Included entire `ThrusterERC20` code in this file |

| Name | Fork | Yield | Changes |
|---|---|---|---|
| ThrusterERC20.sol | UniswapV2ERC20.sol | False | 1. Only name changes to LP token name to `T-LP` and `Thruster LP`<br>2. Changed “Uniswap V2” to “Thruster”<br>3. Removed this file and put the entire code of the file in `ThrusterPair` so that `Transfer` events could be emitted from the factory for easier event tracking |
| ThrusterRouter.sol | UniswapV2Router02.sol | True | 1. Only modified from “UniswapV2” to “Thruster” |
| ThrusterLibrary.sol | UniswapV2Library.sol | False | 1. Only modified from “UniswapV2” to “Thruster” |

# CLMM - Uniswap V3 Changes

- `POOL_INIT_HASH_CODE` = `0x335b9b045380e3536ac34063cce514671792b9cec453f4e92b33f2e0d8c5a770`
- Does not use `ThrusterYield` from V2 due to contract size issues.
- Unless explicitly mentioned, contract names and abis are consistent with Uniswap V3 contracts, both core and periphery.
- Changed all contracts and interfaces with `UniswapV3` to `Thruster`, e.g. `IUniswapV3PoolActions.sol` → `ThrusterPoolActions.sol`
  - The one exception to this is `UniswapV3Factory.sol` → `ThrusterPoolFactory.sol`. From above, you can see the V2 style factory is just `ThrusterFactory.sol`. This is to distinguish V2 from V3.
- One significant change is the separation of the `PoolDeployer` from the `PoolFactory` contract. Therefore, for all contracts that use `PeripheryImmutableState` the `_factory` address should actually be the `_deployer` address, as the contract is deployed by the deployer address storage.
  - This separation was necessary due to contract sizes going over the limit with the addition of V3 gauges and Blast yield additions
  - `PeripheryImmutableState` is usually used for leveraging the `PoolAddress` library to compute the address of the pool through `computeAddress`

**V3 Changelogs**

| Name | Fork | Gas | Yield | Changes |
|---|---|---|---|---|
| ThrusterPool.sol | UniswapV3Pool.sol | True | True | 1. Added `gauge` state variable<br>2. Added calls for `checkpoint` and `cross` ticks for gauge in `swap` for handling V3 gauges, functions similarly to how fees are counted. Should not affect swap code at all besides costing a bit more gas<br>3. Remove `_blockTimestamp()` internal function to reduce contract size |

| Name | Fork | Gas | Yield | Changes |
|---|---|---|---|---|
| | | | | 4. Removed `flash`, flash loans through Uniswap V3 pools due to contract size issues<br>5. Has direct code to claim yield instead of using `ThrusterYield` due to contract size<br>6. Added a call for the Factory to emit all swap events through the pool — helps with tracking swaps through a single address rather than tracking all pool addresses |
| ThrusterPoolDeployer.sol | UniswapV3PoolDeployer.sol | False | False | 1. Separated this contract's storage from Factory due to size limits.<br>2. Set factory as an immutable variable, and require that all `deploy` calls must come from the `factory`<br>3. Removed `factory` variable from `deploy` function signature, as it's in storage now |
| ThrusterPoolFactory.sol | UniswapV3Factory.sol | True | False | 1. Separated this contract's storage from Deployer due to size limits. |
| SwapRouter.sol | SwapRouter.sol | True | False | 1. No changes |
| NonfungiblePositionManager.sol | NonfungiblePositionManager.sol | True | False | 1. Initializes `factory` variable with `ThursterPoolDeployer` address<br>2. Adjusted `IncreaseLiquidity`, `Collect`, and `DecreaseLiquidity` events to also emit ticks and pool address.<br>3. Scoped code in `mint` function to avoid stack too deep |

| Name | Fork | Gas | Yield | Changes |
|---|---|---|---|---|
| NonfungibleTokenPositionDescriptor.sol | NonfungibleTokenPositionDescriptor.sol | False | False | 1. Removed hardcoded mainnet token addresses and added USDB |