

Pair Token Hook

Full range pair LP tokens: one plain ERC-20 per pair (WETHNVDA, WBTCWETH, NVDAUSDG) that deposits into the pair's 1% full range position, auto compounds and redeems pro rata. Permissionless factory plus a Hookr preset launcher. Programmable through the hook, unlike a fixed Uniswap v2 LP token.

 In partnership with Hookr (hookr.fun) for Uniswap v4 hook launches

Date: 2026-10-14
Repository commit: 2b08a46067d5 | source hash: f7bb22ac42ac
Compiler: solc 0.8.26, cancan, via IR (Foundry)
Tests: 19 unit, 7 fork
Publisher: SB (BVI) Ltd, StonkBrokers
Contact: simple@clutch.market, Telegram simplefarmer69

Status: LIVE on Robinhood Chain 4663 (launcher redeployed 2026-10-13, HK-22 fix)

PairTokenFactory 0x282c4815F30abc1D0A8eB810c7309140d35a5B38, PairTokenFeeManager
0x18AdF7be1d23022FcCf286D3d342d92Ecb3F9195, PairToken implementation
0x56C8c249d4c021e93635b120aC68A9a36eb01753, PairTokenLauncher 0x1cEE10c87D261912A833E000f7e7295DC9E813E7,
StonkFeeHook 0x261a19357c66349BccEC55711F81612C0030aaCC. Owner is the StonkBrokers ops deployer
0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C until the company signals the handover. Fee carve: 10% of compounded fees,
split 50 / 50 between the Clock In 3.0 broker engine 0xB673276C46dE1428EacC15Ed6eceF480259D8566 and the STONKBROKER
buyback wallet 0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8; the Hookr Safe
0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 carries partnerBps 5000 on the launcher (half of the StonkBrokers controller
leg). Seed gate: 13 owner listed anchors (ETH, WETH, USDG, STONKBROKER and the ten Clock In stock tokens) with minimum
seed sizes; any pair whose two tokens meet their anchor floors can be created by anyone. On chain tickers are the uppercase
currency order concat (WETHNVDA, USDGNVDA). Unit suite 155/155 + fork 7/7 against the live PoolManager, Sourcify exact match
on all five contracts.

Source and ABIs

Contracts live under hooks/ in the StonkBrokers repository. Byte exported ABIs ship beside this guide under
contracts/hooks/integration/ and are served from stonkbroskers.io/integration/hooks/.

Deployed bytecode sizes

Contract	Role	Runtime size
PairTokenFactory	Create pairs, guards	n/a
PairToken	Per pair share ERC-20 vault	n/a
PairTokenFeeManager	10% carve, booster / buyback / partner	n/a
PairTokenLauncher	Hookr preset launcher	n/a
StonkFeeHook	Shared fee / lock hook	n/a

Contents

- 1. What the hook does
- 2. Why this is not an owned root on Hookr
- 3. Addresses
- 4. Hook identity and hooklist entry

- 5. Pool keys
- 6. Trading the pair token and the underlying
- 7. Contract functions
- 8. Proofs
- 9. Go live status

1. What the hook does

A pair token is an ERC-20 whose supply represents a share of ONE full range Uniswap v4 position on ONE pool. Anyone can create the token for a pair of ERC-20s (or native ETH and an ERC-20) through `PairTokenFactory.create`. The factory derives the hookless `PoolKey` at fee 10000 (1%) and `tickSpacing` 200, initializes the pool if it does not exist yet, clones a `PairToken` behind an EIP-1167 proxy, and routes the creator's seed through the first deposit in the same transaction. The token name is the two tickers concatenated in currency order, wrapped token style: WETH plus NVDA becomes WETHNVDA, WBTC plus WETH becomes WBTCWETH, NVDA plus USDG becomes NVDAUSDG. Each pair has exactly one token (`PairExists` on a second create) and every pair runs on the 1% pool.

Holding the token is holding the LP. The position lives directly in the `PoolManager` under the `PairToken`'s own address (the vault runs its own `unlockCallback`, no `PositionManager` NFT), so the token is the only claim on it. Holders can:

- deposit both sides at the pool ratio (deposit) or one side only (`zapDeposit`, with ETH or either token; the vault swaps part of it through the pool to reach the ratio),
- trade the ERC-20 anywhere an ERC-20 trades, LP it, lock it, send it to a broker ERC-6551 wallet,
- redeem pro rata for both underlying tokens (redeem) or for a single token (`redeemZap`, swapping the other side out through the pool),
- call `compound()` permissionlessly to collect the position's accrued swap fees and fold them back into liquidity. Every deposit and redeem compounds first as well, so share pricing is always on a fee inclusive basket.

Design goals:

- One token per pair, one pool tier (1%), one full range position: no band management, no rebalancing, no impermanent loss realization until a holder redeems.
- Permissionless and free creation on hookless pools: the only requirement is a genesis deposit worth at least the per token minimum seed (`minSeed(token)`, owner settable per token). At least one side of the pair must be an anchor token with a minimum seed set, otherwise create reverts `SeedTooSmall`. Live anchors (13): native ETH 0.004, WETH 0.004, USDG 10, STONKBROKER 2,000, and the ten Clock In 3.0 menu stocks AAPL 0.03, AMZN 0.04, NVDA 0.04, GOOGL 0.03, SPCX 0.06, USAR 0.75, GME 0.4, TSLA 0.025, PLTR 0.05, AMD 0.015 (whole token units, each about 10 USD at the 2026-10-12 marks). The owner extends the list with `setMinSeed`; a USD value floor on the seed is a V2 item.
- Share pricing by basket: `previewDeposit / previewRedeem` quote against `basket()` (position reserves at the live sqrt price plus idle balances). 1,000 dead shares are minted to `0x...dEaD` at genesis so the first depositor cannot inflate share price.
- Churn protection: a 10 bps retention (`RETENTION_BPS`) stays in the vault on every redeem and accrues to remaining holders.
- Zap guards: a single sided deposit may swap at most `maxZapBps` (live 500 = 5% of pool liquidity value) and may not move the pool price more than `maxZapTickMove` ticks (live 200); larger zaps revert `ZapTooLarge / ZapMovesPrice` and the integrator falls back to a paired deposit. Both knobs are owner settable up to the hard caps `MAX_ZAP_BPS` 2000 / `MAX_ZAP_TICK_MOVE` 2000.
- Genesis guard: the first deposit must mint real liquidity. If the pool already exists at a price where the seed adds zero liquidity (an initialized pool at the minimum sqrt price, for instance), create reverts `BadGenesis` instead of minting unbacked shares. The live hookless USDG / NVDA 1% pool on 4663 sits in exactly that state, so a USDGNVDA pair token needs a seed at a real price through a pool that is repriced first.
- The Hookr preset: `PairTokenLauncher.launch` mints a NEW token's first market as a pair token behind a `StonkFeeHook` instance (fee + decaying snipe tax), seeds through the vault, hands the creator the share token, and allowlists the vault as the ONLY LP on that pool so no outside position can dilute the launch LP.

Threat model: Robinhood Chain has no public mempool (private FCFS sequencer), so the guards defend against thin pool bad fills, stale ratio deposits and share price manipulation, not against sandwiching. Every state changing entry is behind a reentrancy lock, every swap and liquidity change passes a caller supplied minimum (`minShares`, `min0 / min1`, `minOut`), and the compound curve can only reach the fee manager, never the owner.

What makes this different from a Uniswap v2 LP token

A Uniswap v2 LP token is a fixed receipt for a fixed pool: one fee tier, no rules, fees sit in the pool until the holder removes liquidity by hand. A pair token is programmable because the rules of its market live in the hook and in the fee manager, not in

the token:

- Pool logic: launcher pairs sit on a v4 pool whose StonkFeeHook sets the swap fee, runs the decaying snipe tax, carves the partner and protocol legs and allowlists the vault as the only LP. A v2 pool has none of these levers.
- Fee routing: the 10% compound carve goes to PairTokenFeeManager, which splits it between the Clock In engine and the buyback wallet. Where the carve lands is configuration, not bytecode on the token.
- Auto compounding: `_compound()` runs on every deposit, zap and redeem, so the position regrows itself and NAV per share only rises. A v2 holder has to collect and re add by hand.
- Single asset entry and exit: `zapDeposit` and `redeemZap` move one asset in or out under the ratio, size and tick guards described above.
- One canonical ERC-20 per pair: the factory refuses a second token for the same two assets, so integrators resolve one address per pair with `pairTokenFor`.
- Plain ERC-20 surface: the share token itself stays a standard ERC-20 that trades on any venue, so wallets and aggregators need no special handling.

2. Why this is not an owned root on Hookr

Factory pair tokens sit on HOOKLESS pools (hooks = 0x0). They are routed with the plain Uniswap v4 Universal Router and the V4 Quoter, not HookrRouter / HookrQuoter. The launcher preset pools carry the launcher's StonkFeeHook instance, which is a StonkBrokers authored base with the Hookr partner leg built in, not a Hookr RulesLaunch root. Neither shape inherits Hookr's own fee blocks; the Hookr share is paid through the partner leg described in section 3.1.

3. Addresses

Item	Chain	Address
PairTokenFactory	Robinhood 4663	0x282c4815F30abc1D0A8eB810c7309140d35a5B38
PairToken implementation (clone target)	Robinhood 4663	0x56C8c249d4c021e93635b120aC68A9a36eb01753
PairTokenFeeManager	Robinhood 4663	0x18AdF7be1d23022FcCf286D3d342d92Ecb3F9195
PairTokenLauncher	Robinhood 4663	0x1cEE10c87D261912A833E000f7e7295DC9E813E7
Launcher StonkFeeHook (flags 0x2acc)	Robinhood 4663	0x261a19357c66349BccEC55711F81612C0030aaCC
Uniswap v4 PoolManager	Robinhood 4663	0x8366a39CC670B4001A1121B8F6A443A643e40951
Uniswap V4 Quoter	Robinhood 4663	0x8dc178efb8111bb0973dd9d722ebef267c98f94
Universal Router 2.1.1	Robinhood 4663	0x8876789976dEcBfCbBbe364623C63652db8C0904
Permit2	Robinhood 4663	0x000000000022D473030F116dDEE9F6B43aC78BA3
WETH	Robinhood 4663	0x0Bd7D308f8E1639FAB988df18A8011f41EAcAD73
Clock In 3.0 broker engine (booster fee leg)	Robinhood 4663	0xB673276C46dE1428EacC15Ed6eceF480259D8566
Hookr partner recipient (launcher preset, 5000 bps of the protocol leg)	Robinhood 4663	0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9
STONKBROKER buyback wallet (fee leg)	Robinhood 4663	0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8
Protocol fee recipient (launcher preset)	Robinhood 4663	0xb668382cF44038a3E8140E789060F6A809787CDa
Owner / deployer	Robinhood 4663	0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C

Deploy parameters (live, state file hooks/deployments/pair-token-hook.4663.json): PoolManager 0x8366a39CC670B4001A1121B8F6A443A643e40951, OWNER 0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C (the ops deployer keeps ownership of factory, fee manager and launcher until the StonkBrokers operator signals a handover), BOOSTER 0xB673276C46dE1428EacC15Ed6eceF480259D8566 (Clock In 3.0 broker engine), BUYBACK 0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8, FACTORY_FEE_BPS 1000 (10% of compounded fee revenue, max 2000), 13 anchor minimum seeds as listed in section 1, PROTOCOL_TO 0xb668382cF44038a3E8140E789060F6A809787CDa, PROTOCOL_BPS 1000, MIN_SEED_QUOTE 0.05 ether, PARTNER_BPS 5000 with PARTNER_TO 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 (Hookr Safe) already set on the launcher. Owner set zap guards: maxZapBps 500, maxZapTickMove 200. The hook is mined through the Arachnid

CREATE2 deployer 0x4e59b44847b379578588920cA78FbF26c0B4956C for flag word 0x2acc.

Deployed size: PairToken 16,211 bytes, PairTokenFactory 7,670 bytes, PairTokenFeeManager 3,647 bytes, PairTokenLauncher 10,377 bytes (EIP-170 limit 24,576).

3.1 Hookr fee leg

Two fee legs exist, one per shape.

Factory pair tokens (hookless pools): there is no swap fee carve, LPs keep the full 1% pool fee. The protocol fee is a performance carve at compound: protocolFeeBps (1000 = 10%) of the fees collected by compound() is pushed to PairTokenFeeManager.notify(token, amount, poolId). Inside the fee manager a per pool Partner record (partnerOf[poolId], to + bps, bps bounded by MAX_PARTNER_BPS 5000) is carved first into accruedPartner[poolId][token]; the remainder lands in accruedProtocol[token]. Permissionless flush(token) pays the protocol accrual 50 / 50 to the Clock In booster and the STONKBROKER buyback wallet; permissionless flushPartner(poolId, token) pays the partner accrual to its recipient. setPartner(poolId, to, bps) is signed by the fee manager owner or the factory owner, so Hookr's address can be attached pool by pool (or to every preset pool) with no redeploy. Until it is set, the partner cut stays in accruedProtocol and nobody can redirect it elsewhere.

Launcher preset pools (StonkFeeHook): the hook's swap fee split applies exactly as on Battle / Strategy / Fractional. protocolBps (1000 = 10%) of each swap fee is the protocol leg, partnerBps (5000 = half of the protocol leg) is carved for Hookr and taken straight to partnerTo 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 (PartnerFeeTaken); it was set at deploy, so no accrual step is needed on the launcher. The rest of the protocol leg goes to protocolTo 0xb668382cF44038a3E8140E789060F6A809787CDa; the controller remainder is also forwarded to protocolTo (FeeForwarded) with a tolerant push, never to the launch creator, whose economics are the pair token shares and the compounded LP fees. The compound carve of section 3.1 above applies to the preset pair token as well.

4. Hook identity and hooklist entry

Factory pair tokens are hookless; list them as vault tokens on vanilla v4 pools. The launcher preset pools carry the launcher's StonkFeeHook instance with flags 0x2acc (beforeInitialize, beforeAddLiquidity, beforeRemoveLiquidity, beforeSwap, afterSwap, beforeSwapReturnsDelta, afterSwapReturnsDelta).

```
{
  "name": "Pair Token Hook",
  "author": "StonkBrokers",
  "address": "0x261a19357c66349BccEC55711F81612C0030aaCC",
  "factory": "0x282c4815F30abc1D0A8eB810c7309140d35a5B38",
  "chainId": 4663,
  "dynamicFee": false,
  "upgradeable": false,
  "requiresCustomSwapData": false,
  "vanillaSwap": true,
  "beforeSwapReturnsDelta": true,
  "afterSwapReturnsDelta": true,
  "source": "https://github.com/Clutch-L4bs/stonkbrosers/tree/main/hooks",
  "docs": "PairToken-Hookr-Integration.pdf",
  "tags": ["full-range", "lp-token", "auto-compound", "pair"]
}
```

5. Pool keys

Shape	currency0 / currency1	fee	tickSpacing	hooks
Factory pair token	the two tokens sorted by address (native ETH = 0x0 sorts first)	10000	200	0x0 (hookless)
Launcher preset	launched token and quote sorted by address	10000	200	launcher StonkFeeHook

Pool id = keccak256(abi.encode(PoolKey)). pairTokenForKey(key) and pairTokenFor(tokenA, tokenB) resolve the pair token; isPairToken(addr) confirms a candidate address was cloned by the factory. PairCreated(pairToken, token0, token1, poolId, creator, hooked) is the discovery event.

6. Trading the pair token and the underlying

The pair token is a plain ERC-20: list it, trade it, LP it like any other token. Its fair value is `previewRedeem(1e18)` priced at the quote asset; UIs should show `basket()` (`reserve0`, `reserve1`) and `positionLiquidity()` beside the token price so holders see the backing.

Trading the UNDERLYING pool (factory shape) is a vanilla v4 swap: quote with the Uniswap V4 Quoter `quoteExactInputSingle` (`hookData` empty), trade with Universal Router V4_SWAP (SWAP_EXACT_IN_SINGLE, SETTLE_ALL, TAKE_ALL), approvals through Permit2. Trading a launcher preset pool follows the `StonkFeeHook` rules: exact input only while the snipe tax is nonzero (`ExactOutputWhileTaxed`), read `currentSnipeBps(id)` first, outside `addLiquidity` / `removeLiquidity` revert `LiquidityLocked` (only the pair token vault may LP).

Entering and exiting the LP goes through the vault, never through the router:

- `deposit(amount0, amount1, minShares, to)`: pull both sides, compound, mint shares. Send native value when `currency0` is ETH (`NativeMismatch` otherwise).
- `zapDeposit(tokenIn, amountIn, minShares, to)`: pull one side (or `msg.value` for ETH), swap toward the ratio inside the zap guards, mint shares.
- `redeem(shares, min0, min1, to)`: burn shares, pull the pro rata slice of the position plus idle, 10 bps stays in the vault.
- `redeemZap(shares, tokenOut, minOut, to)`: redeem then swap the other side into `tokenOut`.

7. Contract functions

7.1 PairTokenFactory

Function	Who	What it does
<code>create(tokenA, tokenB, amountA, amountB, sqrtPriceX96IfNew, minShares)</code> payable	anyone	Create the pair token for a hookless 1% pool, initialize the pool if new (zero <code>sqrtPriceX96IfNew</code> derives it from the seed amounts, else <code>PriceRequired</code>), route the seed as the genesis deposit, mint shares to the creator. Reverts <code>SameToken</code> , <code>PairExists</code> , <code>SeedTooSmall</code> .
<code>createForKey(key, symbol0, symbol1)</code>	allowed creators / owner	Create a pair token on an arbitrary already initialized <code>PoolKey</code> (hooked pools allowed, used by <code>PairTokenLauncher</code>). Reverts <code>NotAllowedCreator</code> , <code>PoolNotInitialized</code> .
<code>pairTokenFor(tokenA, tokenB)</code> view	anyone	Pair token for an unordered token pair (hookless 1% pool).
<code>pairTokenForKey(key)</code> view	anyone	Pair token for an exact <code>PoolKey</code> .
<code>pairTokenOf(token0, token1)</code> view	anyone	Mapping read in currency order.
<code>isPairToken(addr)</code> view	anyone	True when the address is a factory clone.
<code>allPairsLength()</code> view	anyone	Number of pair tokens created.
<code>implementation</code> / <code>poolManager</code> / <code>feeManager</code> / <code>protocolFeeBps</code> / <code>maxZapBps</code> / <code>maxZapTickMove</code> / <code>minSeed(token)</code> / <code>allowedCreator(addr)</code> view	anyone	Configuration reads.
<code>setMinSeed(token, amount)</code>	owner	Per token genesis minimum (<code>address(0)</code> = native).
<code>setAllowedCreator(creator, bool)</code>	owner	Allowlist for <code>createForKey</code> .
<code>setZapGuards(maxZapBps, maxZapTickMove)</code>	owner	Caps <code>MAX_ZAP_BPS</code> 2000, <code>MAX_ZAP_TICK_MOVE</code> 2000.
<code>setProtocolFeeBps(bps)</code>	owner	Compound carve, cap <code>MAX_PROTOCOL_FEE_BPS</code> 2000.

Events: `PairCreated(pairToken, token0, token1, poolId, creator, hooked)`, `MinSeedSet`, `AllowedCreatorSet`, `ZapGuardsSet(maxZapBps, maxZapTickMove)`, `ProtocolFeeSet(bps)`.

7.2 PairToken (one clone per pair)

Function	Who	What it does
<code>deposit(amount0, amount1, minShares, to)</code> payable	anyone	Paired deposit at the live ratio; compounds first; mints shares (<code>MinShares</code> guard).

Function	Who	What it does
zapDeposit(tokenIn, amountIn, minShares, to) payable	anyone	Single sided deposit with ETH or either token; swaps toward the ratio inside the zap guards (ZapTooLarge, ZapMovesPrice).
redeem(shares, min0, min1, to)	holder	Burn shares for both tokens pro rata, 10 bps retention stays for remaining holders (MinOut guard).
redeemZap(shares, tokenOut, minOut, to)	holder	Redeem then swap the other side into tokenOut (BadToken if tokenOut is not in the pair).
compound() returns (fees0, fees1)	anyone	Collect accrued swap fees, carve protocolFeeBps to the fee manager, fold the rest into liquidity. Emits Compounded.
previewDeposit(amount0, amount1) view	anyone	Shares a paired deposit would mint against the current basket.
previewRedeem(shares) view	anyone	(amount0, amount1) a redeem pays after retention.
basket() view	anyone	(reserve0, reserve1): position reserves at the live sqrt price plus idle balances.
positionLiquidity() / tickLower() / tickUpper() view	anyone	Full range position reads.
poolKey() / token0() / token1() view	anyone	Pool identity.
approve / transfer / transferFrom / balanceOf / totalSupply / name / symbol / decimals (18)	anyone	Standard ERC-20.
initialize(poolManager, key, name, symbol)	factory	One shot clone init (AlreadyInitialized, NotFactory).
unlockCallback(bytes)	PoolManager	v4 unlock callback, rejects every other caller (NotPoolManager).

Events: Deposit(sender, to, amount0, amount1, shares), Redeem(sender, to, amount0, amount1, shares), Compounded(fees0, fees1, protocol0, protocol1, liquidityAdded), ERC-20 Transfer / Approval. Constants: DEAD_SHARES 1000 (minted to 0x...dEaD at genesis), RETENTION_BPS 10, DUST 1000.

7.3 PairTokenFeeManager

Function	Who	What it does
notify(token, amount, poolId) payable	pair tokens only	Receive a compound carve, split the per pool partner share into accruedPartner, the rest into accruedProtocol (NotPairToken for anyone else).
flush(token) returns (toBooster, toBuyback)	anyone	Pay accruedProtocol[token] 50 / 50 to booster and buyback.
flushPartner(poolId, token) returns (paid)	anyone	Pay accruedPartner[poolId][token] to the pool's partner recipient.
setPartner(poolId, to, bps)	owner or factory owner	Attach the Hookr recipient to a pool, bps up to MAX_PARTNER_BPS 5000 (PartnerBpsTooHigh, NotFactoryOwner).
setRecipients(booster, buyback)	owner	Rotate the StonkBrokers legs.
setFactory(factory)	owner	One shot wiring (FactoryAlreadySet).
partnerOf(poolId) / accruedProtocol(token) / accruedPartner(poolId, token) / booster / buyback / factory view	anyone	Ledger reads.

Events: FactorySet, RecipientsSet, PartnerSet(poolId, to, bps), FeeNotified(poolId, token, amount, partnerCut), Flushed(token, toBooster, toBuyback), PartnerFlushed(poolId, token, to, amount).

7.4 PairTokenLauncher (Hookr preset)

Function	Who	What it does
----------	-----	--------------

Function	Who	What it does
launch(token, quote, tokenAmount, quoteAmount, buyFeeBps, sellFeeBps, snipeBps, snipeDecaySecs, salt) returns (id, pairToken)	anyone	Initialize the hooked 1% pool, create the pair token through the factory, seed it through the vault, mint the share token to the creator, allowlist the vault as the only LP, start the decaying snipe tax. Reverts HookMismatch (hook unset), NativeNotSupported (zero address side), SeedTooSmall (quote under minSeedQuote), BadParams (fees over MAX_FEE_BPS 1000, snipe over MAX_SNIPE_BPS 9900, decay over 6 hours).
launchInfo(id) view	anyone	Launch struct: creator, token, quote, pairToken, pool, launchedAt, fee terms, totalFees.
currentSnipeBps(id) view	anyone	Live decaying snipe tax.
feeTerms(poolId) view	hook / anyone	Fee terms the hook applies to a pool.
onSwap(...)	hook only	Fee accounting callback from StonkFeeHook; forwards the controller remainder to protocolTo (FeeForwarded).
launchCount / launchOf(poolId) / factory / protocolTo / protocolBps / minSeedQuote / hook / partnerTo / partnerBps view	anyone	Reads.
setHook(hook)	owner	Pins the launcher's StonkFeeHook once.
setPartner(to, bps)	owner	Hookr partner recipient and share of the protocol leg.
setProtocol(to, bps)	owner	Protocol leg recipient / bps (max MAX_PROTOCOL_BPS 2000).
setMinSeed(amount)	owner	Minimum quote seed per launch.

Events: PairLaunched(id, creator, pairToken, token, quote, pool, tokenAmount, quoteAmount, shares), FeeForwarded(id, amount, paid), ProtocolSet, MinSeedSet, PartnerSet(to, bps).

7.5 StonkFeeHook (launcher instance)

Function	Who	What it does
partnerAccrued(quote) view	anyone	Hookr share accrued while partnerTo was unset.
flushPartner(quote) returns (to, amount)	anyone	Release the accrued Hookr share once the launcher's partnerTo is set (PartnerUnset otherwise).
controller() / poolManager() view	anyone	Wiring reads.
lpAllowed(poolId, addr) view	anyone	True for the pair token vault, false for everyone else (LiquidityLocked on outside LP).

8. Proofs

- hooks/test/PairToken.t.sol: 17 test functions, green. Covers permissionless create with pool initialization, the BadGenesis guard on an existing zero liquidity pool, PairExists on a second create, dead shares + genesis minimum, paired and single sided deposits, zap guards (ZapTooLarge / ZapMovesPrice), pro rata redeem with retention, redeemZap, lazy compound folding fees into liquidity with the protocolFeeBps carve reaching the fee manager, fee manager partner carve + flush split, launcher launch minting the share token to the creator, third party LP blocked while the pair token vault is allowed, and the swap fee partner leg on the launcher hook.
- hooks/test/PairToken.fork.t.sol: 7 / 7 green on a Robinhood Chain 4663 fork against the live PoolManager, real WETH and real stock tokens (create, deposit, zap, redeem, compound, launcher preset, genesis guard on the live USDG / NVDA pool).
- Whole hooks suite: 153 / 153 green non fork across Battle, Fractional, Pair Token, Partner, Smart LP Hook, Strategy, plus 7 / 7 fork (160 total) on the deployed source.
- Deposit legs are passed in currency order (token0() / token1() of the pair token), not in the order the creator named them.
- Sizes: PairToken 16,211 B, PairTokenFactory 7,670 B, PairTokenFeeManager 3,647 B, PairTokenLauncher 10,377 B, all under EIP-170.
- Deploy rail: hooks/script/PairTokenDeploy.s.sol (PHASE=deploy / PHASE=verify, HookMiner for flag word 0x2acc, wiring readbacks) broadcast 2026-10-12 and read back clean; Sourcify exact match for all five contracts.

9. Go live status

1. LIVE since 2026-10-12 on Robinhood Chain 4663. Deploy transactions: PairTokenFeeManager 0xa217627a5162ac4488dba0254851bea48691fd1eee994adc9024f52d7d47c2cb, PairTokenFactory 0xee2e5fd173c9adada2a467ed8097601f6b34c6852bc6ec6e87c54cc8e6ecc3fc, PairTokenLauncher 0x4fd0039508e4d375d8fe8ba4e80744b99cbf787d23547097ab3280a55dc1c425, launcher StonkFeeHook (CREATE2) 0x21afc64f2d1098d9be5dcb44f6e9d8d6075b34274f759f75b3f1f353635b92b9, then feeManager.setFactory, launcher.setHook, launcher.setPartner(Hookr Safe, 5000), factory.setAllowedCreator(launcher) and 13 setMinSeed anchor transactions. Full ledger with every hash: contracts/hooks/2026-10-12-pair-token-go-live.json.
2. Verification of record: Sourcify exact match for PairTokenFactory, PairToken implementation, PairTokenFeeManager, PairTokenLauncher and the launcher StonkFeeHook.
3. Ownership: the ops deployer 0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C stays owner of factory, fee manager and launcher until the StonkBrokers operator signals a handover. The owner has no path to vault liquidity or holder shares; owner levers are configuration only (anchors, zap guards, fee bps, partner records).
4. Hookr partner leg: PairTokenLauncher.partnerTo is the Hookr Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 at 5000 bps of the protocol leg on every preset pool. For the compound carve, PairTokenFeeManager.setPartner(poolId, 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9, bps) is signed per preset pool as launches land; both are owner signed configuration, no redeploy.
5. Hookr lists the Pair Token Hook card with the StonkBroker badge and the In partnership with Hookr mark on the StonkBrokers side, routes factory pair pools through Universal Router / V4 Quoter, shows basket() backing beside the token price, and links this guide.
6. Third parties may create pair tokens permissionlessly today on any pair that includes an anchor token. Preset launches signed by StonkBrokers ops keys and any ops seeding of a pair pool remain on hold until the operator supplies name, symbol and seed and confirms a publicly tradeable token may exist.