

Battle Token Hook

Fix notice for Hookr

HK-22: partial fill buys were taxed and counted on the specified amount. New BattleVault and StonkFeeHook bytecode is live on Robinhood Chain.



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

Date: 2026-10-13

Repository commit: 810c9aaf668a

Chain: Robinhood Chain (4663), Uniswap v4 PoolManager 0x8366a39CC670B4001A1121B8F6A443A643e40951

Tests on the shipped bytecode: 155 non fork, 7 fork, 0 failed

Verification of record: Sourcify exact_match on both new contracts

Publisher: SB (BVI) Ltd, StonkBrokers

Contact: simple@clutch.market, Telegram simplefarmer69

What Hookr has to change

Swap the Battle addresses and the two ABIs below in the StonkBrokers hook list and in any Battle front end. Taxed buys that cannot fill their whole specified amount now revert with PartialFill(filled, expected), so the Battle buy flow must never pass a price limit. Nothing else in the integration changes. Strategy, Fractional 404, Smart LP hook and the Pair Token launcher keep their current addresses.

Contents

- 1. Summary
- 2. What was wrong
- 3. What changed in the hook
- 4. New addresses and transactions
- 5. Hookr checklist
- 6. Front end guidance for taxed buys
- 7. Other StonkBrokers hooks
- 8. Proofs and verification

1. Summary

While double checking the Battle Token Hook against the five Hookr review items we confirmed one real game integrity issue. The shared StonkFeeHook charged the buy fee and reported the Battle inflow on the amount the trader specified, not on the amount the pool actually filled. A price limited exact input buy that filled almost nothing still paid the full fee and still counted the full specified amount as net inflow for its side of the battle.

The Battle stack had zero launches, so the fix shipped as a clean redeploy of BattleVault and its StonkFeeHook. The previous generation is archived and must not be referenced. Everything Hookr integrated against (ABI shape, flow, fee split, partner leg, controller interface) is unchanged apart from the two addresses, one new custom error and the buy flow rule in section 6.

2. What was wrong

Uniswap v4 exact input swaps take the specified amount as the trader's intent, but the pool only consumes what the price limit and the available liquidity allow. The old hook did its accounting in beforeSwap on params.amountSpecified and never compared that with the realized BalanceDelta in afterSwap.

Reproduced in the unit suite

- Trader specifies a 50 ETH buy with a price limit one tick away from spot.
- The pool fills about 0.0005 ETH of quote before it hits the limit.
- Old hook: the trader pays about 1.5 ETH of fee (3% of 50 ETH) and the Battle side is credited with 50 ETH of net inflow.
- Consequence: a wallet could buy the 24 hour epoch for its side at the cost of the fee alone, with almost no capital at risk in the pool.

Sells were already correct. The sell branch measures the fee and the Battle outflow on the realized quote delta, so a partial sell pays and counts only what came out.

3. What changed in the hook

afterSwap now compares the realized quote delta with the amount the hook skimmed in beforeSwap. A buy must consume exactly specified minus fee. If it does not, the swap reverts with a new custom error and nothing is charged or counted.

```
error PartialFill(uint256 filled, uint256 expected);

// afterSwap, buy branch (exact input, quote in)
quoteIn = uint256(-params.amountSpecified);
fee = (quoteIn * feeBps) / BPS;
int128 qIn = t.quoteIs0 ? delta.amount0() : delta.amount1();
uint256 consumed = qIn < 0 ? uint256(uint128(-qIn)) : 0;
if (consumed != quoteIn - fee) revert PartialFill(consumed, quoteIn - fee);
```

Why a revert and not a refund: the fee is skimmed from the specified (quote) side in beforeSwap through toBeforeSwapDelta, and a v4 afterSwap may only return a delta in the unspecified currency. An over skimmed quote fee cannot be handed back inside the same swap, so failing closed is the only safe shape. Controller swaps (the vault's own epoch settlement) bypass the fee path as before, and PoolController liquidity is full range, so a normal buy without a price limit always fills completely.

4. New addresses and transactions

Contract	New (live)	Previous (archived, do not use)
BattleVault	0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304	0xc849c3683be5006c8E5B8eec4F3f60A9a855D6c8
StonkFeeHook (Battle)	0x6ced3a0048F9f42836a90ee47FFe64645B0bAacc	0x9afEd91316E1da0334668Dedd3a1dac1d45c6aCc
Hook flags	10956	10956

Setting	Value
Pool manager	0x8366a39CC670B4001A1121B8F6A443A643e40951
Protocol leg	1000 bps to 0xb668382cF44038a3E8140E789060F6A809787CDa

Setting	Value
Partner leg (Hookr Safe)	5000 bps of the protocol leg to 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9
Min seed quote	0.05 ETH
Owner today	0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C
Pending owner	0x6D6981459be6EB09C778d98527bCCc1a235bbb26 (company Safe, acceptOwnership pending)

Transactions (Robinhood Chain)

Step	Tx hash
deployVault	0xa29b2376161434ad9b5a8fa0291e4565b312b52125a9fff20a1d0f766e21aa7a
deployHook	0x7e2fd1aed2c736926f57cfa4b57e259952127237cec923430f24afbe50322626
setHook	0x0f51928ec187cb7eca3aea533396947773966f5e588b0c743ed97e1bbdd3ac37
setPartner	0x90041d36e05753554aea77565efd8694da3f8bc8eef75bb7ba5ef77b8b6cb7fb
transferOwnership	0x0a0e99039276d708e717ae743de5f9aad405685fa20da0fa214bc612120dfe0

Open any hash at <https://brokertools.info/tx/<hash>>.

5. Hookr checklist

1. Replace the BattleVault address with 0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304 everywhere it is referenced (hook list, launch modal, pool discovery).
2. Replace the Battle StokFeeHook address with 0x6ced3a0048F9f42836a90ee47FFe64645B0bAacc. Pools created through the vault carry this hook address in their PoolKey.
3. Reload BattleVault.abi.json and StokFeeHook.abi.json from stonkbrosers.io/integration/hooks/. The only ABI delta is the new PartialFill(uint256,uint256) error on StokFeeHook.
4. Map PartialFill to a plain message in the swap UI (section 6).
5. Drop the archived pair from any cache or index. It has no launches and no pools.
6. No action on Strategy, Fractional 404, Smart LP hook or the Pair Token launcher. Their addresses are unchanged (section 7).

6. Front end guidance for taxed buys

Never pass a price limit on a taxed buy

Use the sqrt bounds as the limit (MIN_SQRT_PRICE + 1 for zeroForOne, MAX_SQRT_PRICE - 1 otherwise) and bound the trade with the router's minimum amount out instead. A limit that stops the swap early now reverts with PartialFill(filled, expected) rather than overcharging the trader.

- Exact input buys: amountSpecified negative, quote in. Fill is always complete against the vault's full range liquidity when no price limit binds.
- Sells (token in): unchanged. Fee and Battle outflow are measured on realized quote out, so partial sells keep working.
- Simulate the swap before signing. A PartialFill in simulation means the user set a price limit or the pool is in a state where the trade cannot fill; show 'This buy cannot fill at the chosen limit, remove the price limit and set a minimum received instead'.
- Settlement of an epoch and the sunrise snipe decay are untouched. Net inflow per side now reflects quote that really entered the pool.

7. Other StonkBrokers hooks

Strategy Token Hook, Fractional 404 Hook, the Smart LP hook and the Pair Token launcher share the same StokFeeHook source but run the previous bytecode. On those surfaces a price limited partial buy only overcharges the trader who set the limit; there is no game outcome to influence, so the issue is self inflicted and the live addresses stay as published. A redeploy of those four stacks with the fixed hook is prepared and waits for StonkBrokers approval. Hookr will receive a second notice with new addresses if and when that ships. Until then the same front end rule applies: no price limit on taxed buys, bound with minimum amount out.

8. Proofs and verification

- Regression tests `test_hk22_partialFillBuyReverts` and `test_hk22_partialFillSellStillWorks` in `hooks/test/Battle.t.sol` (Battle suite 22).
- Full non fork suite 155 passed, 0 failed, plus 7 fork tests against the live PoolManager.
- Deployed sizes: BattleVault 20,496 B, StonkFeeHook 5,860 B, BattleToken 1,845 B (all under EIP-170).
- Sourcify `exact_match`: BattleVault (`exact_match`), StonkFeeHook (`exact_match`). Standard JSON is available on request for a RobinScan submission.
- Deploy state: `hooks/deployments/battle-hook.4663.json`; ledger: `contracts/hooks/2026-10-13-battle-partial-fill-redeploy.json`; audit report: Hashlock Certified Audit, StonkBroker Hooks, finding HK-22.
- Full integration guide with the new addresses: stonkbroskers.io/integration/hooks/StonkBrokers-Battle-Token-Hook-Hookr-Integration.pdf