CoW AMM Liquidity

I'm a solver. How do I use CoW AMM liquidity?

CoW AMM orders already appear in the CoW Protocol orderbook, so you're already using its liquidity. However, CoW AMMs allow solvers to specify custom buy and sell amounts, as long as the order preserves or increase the constant product invariant of the token reserves.

CoW AMMs can be treated as a liquidity source akin to Uniswap or Balancer weighted pools with uniform weights. Each CoW AMM is a pair that trades two tokens.

Importantly, surplus for a CoW AMM order is measured differently when computing the solver reward payout.

Indexing CoW AMMs

CoW AMM pairs can be detected by listening to theConditionalOrderCreated events emitted by theComposableCoW instance for the current chain (see theofficial docs for the current deployment addresses).

The order owner is a CoW AMM if itshandler is the contractConstantProduct . Official deployment addresses for the contracts in this repo can be found in the filenetworks.json .

The fieldstaticInput represents the parameters of the AMM, as for example the traded tokens. A description of each parameter can be found in theinstructions on how to create an order.

The AMM reserves are the owner's balance of the two tokens traded by the AMM.

There are a few caveats to listening to these events:

- CoW AMM orders can be cancelled at any time with a call toComposableCoW.remove
- .
- There is no corresponding event onchain, but the current validity state can be tracked withComposableCoW.singleOrders
- .
- While it's strongly discuraged behavior, there's nothing stopping the same CoW AMM from trading multiple pairs.
- A batch can include only a single custom order per AMM address (this is because a solver cancommit
- to only a single order per AMM address).
- Moreover, if the AMM trades two overlapping pairs, settling one order may affect the reserve balance of an unrelated pair after the order is executed.
- Even if an event is emitted, the order may be impossible to settle.
- This can happen for example if the encoding ofstaticInput
- is invalid.

Settling a custom order

You need to choose a valid CoW Swap order with the following restrictions:

- sellToken
- : any token in the pair.
- buyToken
- : the other token in the pair.
- receiver
- : must beRECEIVER_SAME_AS_OWNER
- (zero address).
- sellAmount
- : any value.
- buyAmount
- : any value such that, after trading these exact amounts, the product of the token reserves is no smaller than before trading.
- validTo
- : at most 5 minutes after the block timestamp at execution time.
- appData
- : must be the value specified instaticInput
- .
- feeAmount
- : must be zero.
- kind
- : any value.
- partiallyFillable

- : any value.
- sellTokenBalance
- : must beBALANCE_ERC20
- .
- buyTokenBalance
- : must beBALANCE_ERC20
- .

You also need to compute:

- the order hashhash
- as defined in the libraryGPv2Order
- , and
- the order signature (example code that generates a valid signature isthegetTradeableOrderWithSignature function in the ComposableCoW repo
-).

This order can be included in a batch as any other CoW Protocol orders with three extra conditions:

- One of the pre-interaction must set the commitment by callingConstantProduct.commit(hash)
- .
- Must contain at most one order from the AMM in the same batch.
- One of the post-interactions must reset the commitment by callingConstantProduct.commit(EMPTY_COMMITMENT)
- .

The last step (clearing the commit) is technically not required for the batch to settle successfully, however it makes the settlement overall cheaper, since it resets the storage slot. However, leaving the commit set means that no AMM orders will appear in the orderbook until the commit is reset. Not clearing the commit at the end of the batch is considered slashable behavior.

Surplus

The surplus for a CoW AMM order is measured differently depending on which AMM order is executed.

If picking the default CoW AMM order (that is, it's settled with the empty commitment), then the surplus is computed exactly like any other CoW Swap order.

Otherwise, if we callX (resp.Y) the reserves of sell (resp. buy) token, andx (resp.y) the sell (resp. buy) amount, then the order surplus is computed as:

x (Y + y) surplus = y - --------- . X

Risk profile

The risks for the funds on the AMM are comparable to the risks of depositing the same reserves on a constant-product curve like Uniswap v2.

The AMM relies on price oracle exclusively for generating orders that will plausibly be settled in the current market conditions, but they aren't used to determine whether an order is valid. If a price oracle is compromised or manipulated, the main risk is that the liquidity available on CoW protocol will be used suboptimally by the solvers that aren't aware of the custom semantics of a CoW AMM. Edit this page Previous Integrating your solver Next CoW Swap Widget