

# #

Coinswap

## #

Summary

This document describes the implementation of the [Constant Product Market Maker Model](#)[open in new window](#) token exchange protocol on IRISHub.

Coinswap supports Token to IRIS and Token to Token exchange. The whole exchange process is implemented on-chain. To generate on-chain assets, you can use the multi-asset management function on IRIS Hub.

Before exchange with Coinswap, the market maker needs to deposit the token to the liquidity pool at the current market price (based on the IRIS token), then the exchange rate between the two tokens on IRIS Hub will vary in real-time according to the on-chain trading situation. When the exchange rate in the liquidity pool is inconsistent with that of other platforms, the arbitrageurs can make a profit. They arbitrage price differences between different platforms and make the exchange rate in the liquidity pool closer to the current market exchange rate.

During the exchange process, the 3/1000 handling fee will be deducted and re-added to the liquidity pool as rewards to the market maker. Market makers can retrieve their tokens at any time without a lock-up period. When the run-off situation occurs, it is useful for the market maker to withdraw the deposit token in time to avoid losses. Therefore, the more tokens deposited in the liquidity pool, the stabler the exchange rate change caused by the exchange process, and the more profit the market maker will make.

## #

Concepts

### #

Liquidity Pool

A system account for depositing mortgage tokens with no control over the private key. The account consists of three parts: IRIS, Token, and liquidity securities (as a certificate for the market maker to hold liquidity and can be transferred). Each token (except IRIS) has its own pool of liquidity to calculate the relative price of the two.

### #

Liquidity

Two assets that can be exchanged in the liquidity pool and mortgaged to the liquidity pool can be considered as providing liquidity for the liquidity pool. When the mortgage assets are withdrawn, the fee charged when the users exchange can automatically be obtained.

### #

Maker

Any individual, organization, or institution that mortgages tokens to a liquidity pool.

### #

Maker formula

Use a constant product as the market making formula:  $x * y = k$ ,  $x$  represents the number of  $x$  tokens, and  $y$  represents the number of  $y$  tokens. During the redemption process, the value of  $k$  remains the same, and the value changes only when the market maker increases/decreases the liquidity.

## #

Actions

- Add Liquidity
- In order to obtain the handling fee during the redemption process, market makers can deposit their tokens into the liquidity pool, mainly in two cases:

- - Create Liquidity Pool
- - If there is no liquidity pool of the token in the IRIShub, the market maker needs to mortgage a fixed amount of tokens and IRIS according to the current market conditions. This step is equivalent to initializing the liquidity pool and pricing the token. If the market maker does not price according to the current market, then the arbitrageur finds that there is a difference in the price, and the exchange behavior will occur until the price is close to the current market price. In this process, the relative price of the token is adjusted entirely by market demand.
- - Add Liquidity
- - If there is a liquidity pool of the token in the IRIShub, when the market maker mortgages the token, it is necessary to mortgage the two tokens according to the current liquidity pool exchange rate. When calculating, we take the IRIS token as the benchmark. If the amount of another token that needs to be mortgaged does not match the current liquidity pool's conversion ratio, the transaction will fail. In this way, as far as possible, the market makers are prevented from making market losses due to the existence of arbitrageurs.
- After the mortgage is completed, the system will lock the deposit token and issue a liquidity voucher to the user account, which can also be transferred.
- Swap Token
- When there is a certain pool of liquidity, the user can initiate a redemption transaction according to his own needs. In the redemption process, the 3/1000 fee is deducted from the input token (this parameter can be changed by the governance module). In terms of the classification of transactions, there are two cases in total:
- - Buy Token
- - If the user purchases a fixed amount of tokens, the IRIShub will calculate the amount of another token the user needs to pay, based on the number of tokens purchased and the current inventory of the liquidity pool. If the amount of tokens paid by the user is less than the value calculated by the IRIShub, the transaction fails.
- - Sell Token
- - If the user sells a fixed amount of tokens, the IRIShub will calculate the amount of another token the user receives, based on the number of tokens sold and the current pool of liquidity. If the number of tokens specified by the user is greater than the value calculated by the IRIShub, the transaction fails.
- In both cases, the IRIShub supports Token's redemption of Token, which requires the collateral of both tokens. The system will redeem twice, Token1 --> IRIS, IRIS-->Token2. A 3/1000 handling fee will be charged for each redemption.
- Remove Liquidity
- After the market maker deposits the token to the IRIShub, he receives the liquidity voucher corresponding to the token, which can be exchanged for the mortgage token and obtain the market-making reward. After the liquidity is withdrawn, the same amount of liquidity voucher will be destroyed from the user's account and the pool.

## #

### Additional information

This module does not provide a command entry but the relevant REST interfaces, through which you can initiate the above transactions.