

# Type Conventions

Regardless of command type of callpath Ambient employs a series of standard conventions when representing various parameters.

## Tokens

Tokens are represented as 160-bit Ethereum addresses. For standard ERC20 token types the address of the token tracker.

For the native currency of the chain (e.g. ETH for Ethereum, MATIC for Polygon, AVAX for Polygon), this is always represented as the 0x0 address. Ambient does not use WETH or wrappers on chain native tokens. Users should always use the zero address for ETH of the chain's native token.

Virtual tokens are represented as the trailing 160 bits of the keccak256 hash of the virtual token tracker address and the virtual token salt. Hence virtual tokens are represented in address space, but the address will always be empty.

Pairs are represented as the joint combination of a base token and a quote token. For a given pair, the lower address must always be used as the base token of the pair. (Hence chain native ETH will always be the base token).

## Pool Type Index

An essentially unlimited number of pools can be supported for any given pairs. Pool types map to specific properties of a pool, such as fee rate, tick size and whether the pool is permissioned. Pool type indexes apply consistently across any possible pair.

Pool index is a 256-bit number chosen arbitrarily by protocol governance. Pool type index is fixed for the lifetime of any given pool. For permissioned pools the leading 160-bits of the pool index is the address of the permission oracle.

Currently Ambient only has one pool type index initialized and it is 420

## Token Quantities

Token quantities are represented as either signed or unsigned 128-bit integers. Note that token quantities are raw values, and do not normalize for the ERC20 token decimal value.

Signed token quantities indicate the direction of the flow relative to the pool. Negative token quantities indicate a credit received by the user from the pool. Positive token quantities indicate a debit paid to the pool by the user.

## Prices

All curve prices are represented on chain as fixed-point Q64.64 representations of the square root. Curve prices always represent the instantaneous exchange rate between the raw tokens and do not normalize for the ERC20 token decimals.

The SDK contains helper functions for correctly converting to/from floating point prices to Croc native Q64.64 price root representations.

## Ticks

Ticks are represented as signed 24-bit integers that deterministically map to/from curve prices. The tick value is represented as a function of price (P):

$$\lfloor \log_{1.0001} P \rfloor$$

## Settlement Flags

Settlement flags are a light-weight way to indicate trade two-sided trade settlement. Settlement flags are a single 8-bit integer. Flag values are mapped to the following:

- 0x1
- Indicates that the base side of the pair should be settled with surplus collateral
- 0x2
- Indicates that the quote side of the pair should be settled with surplus collateral
- 

All remaining bits are currently unused, but may be in the future. As such the client should set all other bits to 0 for future backwards compatibility.

## Pool Liquidity Fees

Pool liquidity fees (or "swap fees") are represented as unsigned 16-bit integers, and represent units of 0.0001%.

