

Introduction to Hooks

Traditional trading protocols don't allow the execution of logic immediately before or after a trade. It is then difficult for users of intent-based protocols to handle complex interactions, as they must be carried out over multiple transactions. Some attempts to address this have involved broadcasting different order types to Fulfillers, however, we believe this approach places a significant burden on the Fulfillers, as they must execute bespoke logic in addition to filling orders.

Flood's solution to this problem is the introduction of hooks.

Hooks are mechanisms that allow the execution of arbitrary logic before and after the execution of an order.

They allow composing with other protocols and enable a wide array of actions such as:

- Bridging to a different chain immediately after trade
- Withdrawing from a Lending market before a trade
- Buying ETH and immediately depositing it in a liquid staking protocol
- Any other smart contract logic you can think of!

Hooks can either be pre or post trade. As the name implies, pre-trade are executed first, before tokens are transferred from the offerer to the Fulfiller and before the Zone validates the order. Post-trade hooks execute after a trade has happened and tokens have been transferred to the recipient.

Hooks be signed explicitly by users to generate a valid order signature. This is done to ensure transparency over trade execution and the option to opt out of certain hooks if needed.

Hooks are implemented as target address and calldata, meaning any existing smart contract function can be used as a hook. Flood Plain calls the target of each hook, meaning it will be msg.sender in each hook call context. Any call is allowed in hooks, beside calls starting with theMAGIC_VALUE , which is done for safety reasons.

How To Write Hooks

Start from forking the Flood[Hook repository\(opens in a new tab\)](#).

Here's a few things you should keep in mind when developing hooks:

- msg.sender
- is FloodPlain.
- Pre-hooks are executed before funds are transfered from the offerer.
- Post-hooks are executed after funds are transferred to the recipient.
- Zones and Fulfillers can deem orders with certain hooks invalid.

The Flood team is always available on[Discord\(opens in a new tab\)](#) to guide you through the process!

FAQ

- Question
- : How are Flood hooks different from UniswapV4 hooks?Answer
- : UniswapV4 hooks are set on per pool basis and are immutable. Flood Hooks are arbitrary and can be set on a per-order basis.
- Question
- : What is the difference between pre-trade hooks and smart contract signatures, and when to use each?Answer
- : Pre-hooks are best suited for synchronous logic that executes before a trade, while smart contract signatures are a better fit for reacting to asynchronous action before a trade, usually in a separate transaction.
- Nothing prevents you from combining both into a single smart contract.
- Question
- : Why is the magic value blacklisted in calldata?Answer
- : The magic value is blacklisted to prevent hooks from callingsourceConsideration
- on a Fulfiller and posing as FloodPlain. We blacklist 32 bytes instead of the 4 bytes of the function selector to make collisions with legitimate functions less likely.
- Question
- : I need to know the consideration in my hook, but I won't know it until after the trade happened.Answer
- : You can set the hook as recipient, then use the whole balance (or the change in balance) as the consideration received.

Last updated onFebruary 3, 2024 [Everything About Zones Offchain Infrastructure](#)