Warp Routes: Types

Warp routes are Hyperlane's implementation of token bridging, allowing for permissionless transfer of native, ERC20, and synthetic (newly deployed ERC20) assets across any chain via Hyperlane. This document provides details on all warp route types.

Please note that this document does not include ERC721 warp routes.

Native Token Warp Routes

Implemented inHypNative.sol , native warp routes handle the transfer of native gas tokens (e.g. ETH on Ethereum or Arbitrum, MNT on Mantle) across different chains.

Features

- Directly transfers native tokens without wrapping.
- Usesmsg.value
- for transfer amount.
- Handles excessmsg.value
- as hook payment.
- Supports donations through areceive()
- function.

Seethe implementation for more details.

Collateral-Backed ERC20 Warp Routes

Implemented inHypERC20Collateral.sol , collateral warp routes enable the transfer of ERC20 tokens across chains by locking them as collateral.

Features

- Wraps existing ERC20 tokens as collateral for transfers.
- Locks tokens in the contract on the source chain.
- Releases equivalent tokens on the destination chain.
- Uses SafeERC20 for secure token transfers.

Seethe implementation for more details.

Synthetic ERC20 Warp Routes

Implemented inHypERC20.sol , synthetic warp routes create new tokens on destination chains that represent tokens from the origin chain.

Features

- Maintains consistent total supply across all chains.
- Supports custom token attributes (name, symbol, decimals).
- Mints new tokens on the destination chain.
- Burns tokens on the source chain when transferred back.

Seethe implementation for more details.

TokenRouter Functionality

All warp routes extend theTokenRouter contract, which provides the core functionality for warp route token transfers.

Features

1. Message Structure
2. : UsesTokenMessage
3. library for encoding and decoding token transfer messages.
4. Transfer Initiation
5. :transferRemote
6. function initiates cross-chain transfers.

7. Message Handling
8. :_handle
9. function processes incoming transfer messages.
10. Abstract Methods
11. :* _transferFromSender
12.
    - : Implemented by all warp routes to handle token collection.
13.
    - _transferTo
14.
    - : Implemented by all warp routes to handle token distribution.

TokenMessage Format

[32

bytes

for recipient] [32

bytes

for amount] [remaining bytes

for metadata] This standardized format ensures consistent handling across different warp route implementations while allowing for extensibility through metadata.

Seethe implementation for more details.

FastTokenRouter Transfers

Implemented inFastTokenRouter.sol , this router extends TokenRouter and provides faster token transfers through a liquidity provider mechanism.

Features

- Allows liquidity providers to fulfill transfer requests before message processing.
- Includes afastFee
- to incentivize liquidity providers.
- IntroducesfastTransferId
- for unique transfer identification.

Seethe implementation for more details.

Specialized Warp Route Extensions

1. Fast Collateral Transfers (FastHypERC20Collateral)

Combines fast transfer capabilities with collateral-backed ERC20 functionality. Seethe implementation for more details.

2. Vault Integration (HypERC4626OwnerCollateral, HypERC4626Collateral)

Allows for yield generation on collateral by integrating with ERC-4626 vaults. Seethe implementation &rebasing variant and for more details.

3. Fiat-Backed Tokens (HypFiatToken)

Designed for stablecoins and other fiat-backed tokens, implementing specific mint and burn operations. Seethe implementation for more details.

4. Scaled Native Tokens (HypNativeScaled)

Scales native token values for consistency across chains with different decimals. Seethe implementation for more details.

5. xERC20 Integration (HypXERC20 & HypXERC20Lockbox)

Enables cross-chain transfers of xERC20 tokens, integrating with lockbox mechanisms for conversions. See theHypXERC20 implementation and theHypXERC20Lockbox implementation for more details.

info For setup examples & use cases, check out Warp Routes: Example Usage . Edit this page Previous Warp Route Interface Next Warp Routes: Example Usage