Adapters

Adapters are contracts that can hook into the normal flow of cross-chain transactions and augment their capabilities without requiring changes in existing contracts. Theconnext/integration repository contains adapters that can be inherited for these purposes.

SwapAdapter

This adapter contains the logic for swapping tokens. TheSwapAdapter can be used on either the origin or the destination side to execute a swap.

Using on Origin

SwapAndXCall is a contract that implementsSwapAdapter and is meant to be used on the origin chain. It swaps the input tokens into desired output tokens before initiating the cross-chain transaction withxcall . This is useful in cases where you want users to be able to send any token to your contract and bridge them through Connext.

Using on Destination

SwapForwarderXReceiver also implementsSwapAdapter but it's used on the destination chain. It swaps the tokens received from the bridge into desired output tokens before proceeding with the "forward call", which contains the rest of the logic that follows on the destination side. TheForwarderXReceiver that it implements is detailed in the next section for Receivers.

Swappers

TheSwapAdapter holds a registry ofallowedSwappers which are contracts that implement theISwapper interface:

```
Copy interfaceISwapper{ functionswap( uint256_amountIn, address_tokenIn, address_tokenOut, bytescalldata_swapData )externalpayablereturns(uint256amountOut); }
```

For example, theUniV3Swapper implementsswap which internally calls Uniswap'sISwapRouter.exactInputSingle to execute the swap via Uniswap.

Currently, Connext provides the following Swappers:

- OneInchUniswapV3
- UniV2Swapper
- UniV3Swapper
-

Previous Integration Next Receivers On this page * SwapAdapter * Using on Origin * Using on Destination * Swappers

Edit on GitHub