

USDC Transfers With Connect SDK

The Connect SDK enables fast, cheap, native USDC bridging powered by Circle's Cross Chain Transfer Protocol. Using [CCTP](#) to transfer native USDC across chains with the [Connect SDK](#) works very much like a standard Token Transfer with the SDK.

Installation

First install the required packages to use CCTP on EVM compatible platforms (or others as they're supported)

...

```
Copy npm install @wormhole-foundation/connect-sdk @wormhole-foundation/connect-sdk-evm @wormhole-foundation/connect-sdk-evm-cctp
```

...

Usage

To use the CCTP bridge, the platform must be imported and the protocol must be registered.

...

```
Copy import { Wormhole } from "@wormhole-foundation/connect-sdk"; import { EvmPlatform } from "@wormhole-foundation/connect-sdk-evm";
```

```
// register the protocol by importing the module import "@wormhole-foundation/connect-sdk-evm-cctp";
```

...

With this done, we can create the Wormhole instance that can be used to create a Chain Context object.

...

```
Copy // init Wormhole object, passing config for which network // to use (e.g. Mainnet/Testnet) and what Platforms to support
const wh = new Wormhole("Testnet", [EvmPlatform]);
```

...

Manual Transfer

...

```
Copy // 1.0 USDC in base units const amt = normalizeAmount("1.0", 6)
```

```
const srcAddress = Wormhole.chainAddress("Ethereum", "0xdeadbeef...")
const dstAddress = Wormhole.chainAddress("Avalanche", "0xbeefdead...")
```

```
const xfer = await wh.circleTransfer(amt, srcAddress, dstAddress); console.log(xfer);
```

...

A Manual transfer has 3 steps:

1. Initiate The transfer by calling `initiateTransfer`
2. and passing a [Signer](#)
3. to sign the transactions.
- 4.

...

```
Copy console.log("Starting Transfer"); const srcTxids = await xfer.initiateTransfer(src.signer); console.log(Started Transfer, srcTxids);
```

...

1. Wait for the Circle Attestation to be available, optionally passing a
- 2.

...

```
Copy // See https://developers.circle.com/stablecoins/docs/required-block-confirmations for reasonable timeout settings //
```

based on origin chain consttimeout=60*1000;

```
console.log("Waiting for Attestation"); constattestIds=awaitxfer.fetchAttestation(timeout); console.log(Got Attestation:.,attestIds);
```

...

1. Complete the transfer by calling, you guessed it,completeTransfer
2. and again passing a Signer for the destination chain
- 3.

...

```
Copy console.log("Completing Transfer"); constdstTxids=awaitxfer.completeTransfer(dst.signer); console.log(Completed Transfer:.,dstTxids);
```

...

Automatic Transfer

Using the Automatic Relaying feature is even easier and only involves initiating the transfer. The Relayer infrastructure will handle fetching and delivering the Attestation for you.

...

```
Copy constxfer=awaitwh.circleTransfer( amount, srcAddress, dstAddress, true,// automatic transfer plz undefined,// An arbitrary bytes payload if one is necessary 0n,// no native gas dropoff for this demo ); console.log(xfer);
```

```
console.log("Starting Transfer"); constsrcTxids=awaitxfer.initiateTransfer(src.signer); console.log(Started Transfer:.,srcTxids);
```

...

Complete Partial Transfer

In the case that a manual transfer is started but not finished, the transfer object can be reconstituted from only the source chain and transaction hash.

This is especially useful in cases where a user has terminated their session prior to completing the transfer or even for debugging.

...

```
Copy consttimeout=60*1000
```

```
// Rebuild the transfer from the source txid constxfer=awaitCircleTransfer.from(wh,{ chain:"Avalanche", txid:"0x6b6d5f101a32aa6d2f7bf0bf14d72bfbf76a640e1b2fdbbeeac5b82069cda4dd", },timeout);
```

```
constdstTxids=awaitxfer.completeTransfer(signer); console.log("Completed transfer: ",dstTxids);
```

...

The full source of a working example is available[here](#)

Last updated1 month ago

On this page * [Installation](#) * [Usage](#) * [Manual Transfer](#) * [Automatic Transfer](#) * [Complete Partial Transfer](#)

Was this helpful? [Edit on GitHub](#)