Bridging Your Standard ERC-20 Token Using the Standard Bridge

This tutorial is for developers who want to bridge a new ERC-20 token to an OP Stack chain. If you want to bridge existing tokens, you can skip to the tutorial on Bridging ERC-20 tokens with the Optimism SDK. In this tutorial you'll learn how to bridge a standard ERC-20 token from Ethereum to an OP Stack chain using the Standard Bridge system. This tutorial is meant for developers who already have an existing ERC-20 token on Ethereum and want to create a bridged representation of that token on layer 2.

This tutorial explains how to use the OptimismMintableERC20Factory (opens in a new tab) to deploy a standardized ERC-20 token on layer 2. Tokens created by this factory contract are compatible with the Standard Bridge system and include basic logic for deposits, transfers, and withdrawals. If you want to include specialized logic within your L2 token, see the tutorial on Bridging Your Custom ERC-20 Token Using the Standard Bridge instead.

The Standard Bridge does not support fee on transfer tokens (opens in a new tab) or rebasing tokens (opens in a new tab) because they can cause bridge accounting errors.

About OptimismMintableERC20s

The Standard Bridge system requires that L2 representations of L1 tokens implement the IOptimismMintableERC20 (opens in a new tab) interface. This interface is a superset of the standard ERC-20 interface and includes functions that allow the bridge to properly verify deposits/withdrawals and mint/burn tokens as needed. Your L2 token contract must implement this interface in order to be bridged using the Standard Bridge system. This tutorial will show you how to use the OptimismMintableERC20Factory (opens in a new tab) to create a basic standardized ERC-20 token on layer 2.

Dependencies

- cast (opens in a new tab)

Get ETH on Sepolia and OP Sepolia

This tutorial explains how to create a bridged ERC-20 token on OP Sepolia. You will need to get some ETH on both of these testnets.

You can use this faucet (opens in a new tab) to get ETH on Sepolia. You can use the Superchain Faucet (opens in a new tab) to get ETH on OP Sepolia.

Get an L1 ERC-20 Token Address

You will need an L1 ERC-20 token for this tutorial. If you already have an L1 ERC-20 token deployed on Sepolia, you can skip this step. Otherwise, you can use the testing token located at 0x5589BB8228C07c4e15558875fAf2B859f678d129 (opens in a new tab) that includes a faucet() function that can be used to mint tokens.

Create an L2 ERC-20 Token

Once you have an L1 ERC-20 token, you can use the OptimismMintableERC20Factory (opens in a new tab) to create a corresponding L2 ERC-20 token on OP Sepolia. All tokens created by the factory implement the IOptimismMintableERC20 interface and are compatible with the Standard Bridge system.

Add a private key to your environment

You'll need a private key in order to sign transactions. Set your private key as an environment variable with the export command. Make sure this private key corresponds to an address that has ETH on OP Sepolia.

export TUTORIAL_PRIVATE_KEY = 0 x...

Add an OP Sepolia RPC URL to your environment

You'll need an RPC URL in order to connect to OP Sepolia. Set your RPC URL as an environment variable with the export command.

export TUTORIAL_RPC_URL = https://sepolia.optimism.io

Add your L1 ERC-20 token address to your environment

You'll need to know the address of your L1 ERC-20 token in order to create a bridged representation of it on OP Sepolia. Set your L1 ERC-20 token address as an environment variable with the export command.

Replace this with your L1 ERC-20 token if not using the testing token!

export TUTORIAL_L1_ERC20_ADDRESS = 0x5589BB8228C07c4e15558875fAf2B859f678d129

Deploy your L2 ERC-20 token

You can now deploy our L2 ERC-20 token using the [OptimismMintableERC20Factory (opens in a new tab)](#). Use the cast command to trigger the deployment function on the factory contract. This example command creates a token with the name "My Standard Demo Token" and the symbol "L2TKN". The resulting L2 ERC-20 token address is printed to the console.

cast

send

0x4200000000000000000000000000000000000012

"createOptimismMintableERC20(address,string,string)" TUTORIAL_L1_ERC20_ADDRESS "My Standard Demo Token"

"L2TKN"

--private-key PRIVATE_KEY --rpc-url TUTORIAL_RPC_URL --json

|

jq

-r

'.logs[0].topics[2]'

|

cast

parse-bytes32-address

Bridge Some Tokens

Now that you have an L2 ERC-20 token, you can bridge some tokens from L1 to L2. Check out the tutorial on [Bridging ERC-20 tokens with the Optimism SDK](#) to learn how to bridge your L1 ERC-20 to L2s using the Optimism SDK.

Add to the Superchain Token List

The [Superchain Token List (opens in a new tab)](#) is a common list of tokens deployed on chains within the Optimism Superchain. This list is used by services like the [Optimism Bridge UI (opens in a new tab)](#). If you want your OP Mainnet token to be included in this list, take a look at the [review process and merge criteria (opens in a new tab)](#).

[Bridging Your Custom ERC-20 Token to OP Mainnet](#)[Viewing Deposits and Withdrawals by Address](#)