Detailed Setup Guide

Hello! This section will guide you through the process of setting up yourxERC20 token.

Prerequisites

Let's begin by getting a comprehensive understanding of the required steps based on your token's current situation.

1. Categorize your token

Determine which of the following categories best describes your token's current state.

- Category A
- : The token is new and is not deployed anywhere.
- Category B
- : The token already exists on one chain.
- Category C
- : The token already exists on multiple chains.
-

If you want to spin up anxERC20 on testnet, the category may be different from your mainnet token. We recommend emulating your mainnet token setup if your goal is to testrun the process on testnet.

For example: on mainnet, your tokenTKN is currently only deployed on Ethereum (Category B). Then on testnet, you should deploy aTKN to Goerli and follow the steps for aCategory B token.

1. Define your token's "home" chain

Based on your token's category:

- Category A
- : Choose one chain to be the home chain.
- Category B
- : The chain where your token is currently deployed will be the home chain.
- Category C
- : Choose one chain (among the ones your token is currently deployed) to be the home chain.
-

1. Prepare to deploy tokens

2. Category A

3. :
4.
    - You will deploy anxERC20
5.
    - on each chain you wish to support, including the home chain.
6. *
7. Category B
8. andCategory C
9. :
10.
    - On each chain with an existing token, you need to figure out if a Lockbox setup is needed (next section).
11.
    - On all other chains you wish to support, you will just deploy anxERC20
12.
    - .
13. *
14.

Lockbox Setup

A Lockbox allows any existing ERC20 to become compatible with the ERC-7281 (xERC20) standard. The Lockbox is just a simple wrapper contract, analogous to Wrapped ETH.

ForCategory B andCategory C , there are tokens that already exist on certain chains. For each of these tokens, follow this flowchart to determine if you need to have a Lockbox setup on that token's chain.

Example: NEXT token

To better understand how a Lockbox setup operates, consider theNEXT token as a real-world example.

- On Ethereum: Both theNEXT
- token andxNEXT
- token are deployed with aLockbox
- .
- On Arbitrum: Only thexNEXT
- token is deployed.
-

Bridging from Ethereum to Arbitrum:

- [Ethereum] User depositsNEXT
- into the Lockbox and receivesxNEXT
- [Ethereum] User calls the bridge usingxNEXT
- [Arbitrum] Bridge deliversxNEXT
- to the user
-

Bridging from Arbitrum to Ethereum:

- [Arbitrum] User calls the bridge usingxNEXT
- [Ethereum] Bridge withdrawsNEXT
- from the Lockbox using the bridgedxNEXT
- [Ethereum] Bridge deliversNEXT
- to the user
-

Deploying Contracts

Now that you have an idea of how your tokens should be set up, let's move on to the actual deployment procedures.

xERC20s and Lockboxes

The Wonderland team provides anxERC20 Github repository that contains fully compliant implementations ofxERC20 ,Lockbox , and scripts to deploy them. Factory contracts have already been deployed on each chain listed under/broadcast/MultichainDeploy.sol/{chain_id}/run-latest.json .

We suggest you deploy from a fork of this repo, please see theREADME for instructions. You will configure the scripts based on which chains you need to have Lockbox setups.

If you wish to roll your own version of an xERC20, make sure your custom implementation is compliant with the standard.

TheERC-7281 specification requires compliant tokens to implement ERC-20 along with mint/burn and some additional rate limit interfaces. The absoluteminimal interface needed is the ERC-20 interface plus mint/burn:

```

Copy / *@noticeMints tokens for a user*@devCan only be called by a bridge*@param_user The address of the user who needs tokens minted* @param_amount The amount of tokens being minted */
functionmint(address_user,uint256_amount)external;

/ *@noticeBurns tokens for a user*@devCan only be called by a bridge*@param_user The address of the user who needs tokens burned* @param_amount The amount of tokens being burned */
functionburn(address_user,uint256_amount)external;

```

LockboxAdapter

You might have noticed there's aLockBoxAdapter contract in the diagram above when you have a Lockbox setup. This contract facilitates the unwrapping ofxERC20 -> ERC20 on the destination chain and is needed for UIs to do this step for users.

The Connext team has aLockboxAdapter deployed to all our supported chains (the implementation is availablehere). Use these if you plan to enable Connext as a bridge and want your token listed in our UI.

Mainnets:

- Ethereum: 0x45bf3c737e57b059a5855280ca1adb8e9606ac68

- Polygon: 0x6777c6713F13e499232B3a0CdA246e357a9Cf5EB
- Optimism: 0x81dADc774d2ae44Eb30D2290d076Ae67F9800bd5
- Arbitrum: 0x0B52cA1406eeA3Ce1fcc37dC0121845eF1de3Ae8
- Gnosis: 0x3Cb55bFBB1f4973FCb9705Bab4aBb7E72BF85eAF
- BNB: 0xB71D06f2e73918386B75c24dD26c95DD938f7912
- Linea: 0x4895aa5d666c81a04ebcc7a9aa47f249b1c46aa6
- Mode: 0x981A06E33b228299f33a45a892d7AA61d0b95a83
- Base: 0x542AaB88B14055e47222791276967Db5f9B840f6
- Metis: 0x5bB83e95f63217CDa6aE3D181BA580Ef377D2109
-

Testnets:

- Goerli: 0x8528d3B173a149CB0eae5e2d7335123bcF888c9f
- Optimism-Goerli: 0x55fc076b9b0618a657f8986c83a97bd06e8a80f2
- Mumbai: 0xec531b272fa2b82efe49ef6d830c2d848753642a
-

Whitelisting bridges

As the token issuer, you have the power to decide which bridges can mint/burn your token and the ability to set rate limits per bridge:

```

Copy /* *@notice Updates the limits of any bridge * @dev Can only be called by the owner * @param _mintingLimit The updated minting limit we are setting to the bridge * @param _burningLimit The updated burning limit we are setting to the bridge * @param _bridge The address of the bridge we are setting the limits too*/ function setLimits(address _bridge, uint256 _mintingLimit, uint256 _burningLimit) external;

```

These limits will replenish after_DURATION (by default the repo deploysxERC20 with a value of 1 day).

Once your token is deployed, you can callsetLimits to grant any bridge the privilege to mint/burn your token on that chain.

Enabling Connext as a Bridge

If you want Connext to be able to bridge your token (here's our pitch in the next section), please go through the following steps.

1. Set rate limits for Connext.
2.
   - For each chain where your xERC20 is deployed, callsetLimits
3.
   - as the owner/governor. Use the appropriate Connext address listed under "Core Contracts"here
4.
   - as the_bridge
5.
   - parameter.
6. *
7. Submit a PR to ourChainData mappings
8. .
9.
   - For each chain where your xERC20 is deployed, add an object keyed by its address like this:
10.

    -
      ```
11. 
    - Copy
12. 
    - "0x4c781E4D22cfaAdA520cAe4aF9097C5ecf9C3A71": {
13. 
    - "name":"xDappRadar",
14. 
    - "symbol":"xRADAR",
15. 
    - "decimals":18
16. 
    - }
      ```

17.

    -
      ```
18. *
19. Submit a PR to our[allowlisting scripts]
20. .
21.
    - Under theassets
22.
    - key in the configuration object, add another object to the list like this:
23.
    - ```
24.
    - Copy
25.
    - {
26.
    - name:"RADAR",
27.
    - canonical:{
28.
    - domain:"11111",
29.
    - address:"0x202426c15a18a0e0fE3294415E66421891E2EB7C",
30.
    - decimals:18,
31.
    - },
32.
    - representations:{
33.
    - /// ETHEREUM
34.
    - "6648936":{
35.
    - local:"0x202426c15a18a0e0fE3294415E66421891E2EB7C",
36.
    - adopted:"0x202426c15a18a0e0fE3294415E66421891E2EB7C",
37.
    - },
38.
    - /// BSC
39.
    - "6450786":{
40.
    - local:"0x489580eB70a50515296eF31E8179fF3e77E24965",
41.
    - adopted:"0x489580eB70a50515296eF31E8179fF3e77E24965",
42.
    - },
43.
    - },
44.
    - },
45.
    - ```
46.
    - Thecanonical
47.
    - object should always have11111
48.
    - as thedomain
49.
    - . Change theaddress
50.
    - anddecimals
51.
    - to match your home chain xERC20. For example,RADAR
52.
      ```

- 's home chain is Ethereum.

53.
- For each chain where your xERC20 is deployed (including the home chain), add an entry into therepresentations

54.
- field keyed on the chain'sdomainId

55.
- .

56.
-
  - [You can look up each chain's domainId here]

57.
-
  - . We encourage commenting the chain name above each entry.

58.
-
  - local

59.
-
  - andadopted

60.
-
  - should both be set to the xERC20 address. These exist as separate fields for non-xERC20 assets.

61.
-
  - Note:

62.
-
  - domainId

63.
-
  - is a Connext-specific identifier per chain that exists for forward compatibility with non-evm chains.

64.
- *

65. *
66.

Please reference the ChainData PR in the allowlisting script PR to expedite the review process! Once this is done, the Connext Labs team will review your PRs to sanity check deployment details. Once your PR is approved, your tokens will be whitelisted and transferrable across chains!

Connextscan and Bridge UI Support

To get your token listed on our[Bridge UI] and be able to track transfers in the[Connext explorer] , you'll need to submit two more PRs:

1. Submit a PR to the[Connextscan config]
2. .
3.
   - Create a new object in the list. UsingRADAR
4.
   - as an example where it has a lockbox setup on Ethereum as its home chain:
5.

   -
     ```
6.
   - Copy
7.
   - {
8.
   - "id":"radar",
9.
   - "symbol":"RADAR",
10.
   - "name":"DappRadar",
11.
   - "image":"/logos/assets/radar.png",
12.
   - "is_xERC20":true,
13.
   - "contracts":[
     ```

14.
    - {
15.
    - "contract_address":"0x44709a920fCcF795fbC57BAA433cc3dd53C44DbE",
16.
    - "chain_id":1,
17.
    - "decimals":18,
18.
    - "symbol":"RADAR",
19.
    - "xERC20":"0x202426c15a18a0e0fE3294415E66421891E2EB7C",
20.
    - "lockbox":"0xFf6792A39F44FB67B4796906a5Cb77C677328858",
21.
    - "lockbox_adapter":"0x6ea3dc2e17a0466b36dd3258574e0bd2e4685452"
22.
    - },
23.
    - {
24.
    - "contract_address":"0x489580eB70a50515296eF31E8179fF3e77E24965",
25.
    - "chain_id":56,
26.
    - "decimals":18,
27.
    - "symbol":"RADAR"
28.
    - }
29.
    -]
30.
    - }
31.

    -
      ```
32.
    - For each chain where your xERC20 is deployed, add an object to thecontracts
33.
    - list. If your token has a lockbox setup, thecontract_address
34.
    - should be the ERC20 instead of the xERC20. It should include the other relevant fields shown in the example above.
35. *
36. Submit a similar PR to theBridge UI config
37. . You can use the exact same object here.
38.

And that’s it! Once these PRs are merged, you'll have your token appear onhttps://connextscan.io andhttps://bridge.connext.network .

Router Liquidity

At this point, your token should be transferrable across chains with no added fees or slippage. However, because of how Connext’s model works, these transfers will happen in large batches through Ethereum L1 roughly once every 2-3 hours.

Learn more aboutfast and slow path execution here . If your usecase requires fast (i.e. <2 minute) transfers across chains, you will need some routers in our network to supply some liquidity to execute transactions immediately on behalf of users.

Please reach out to the Connext team and we can help work through options here with our router partners.
      ```