



Get Ahead of the Game with Verus' Groundbreaking 1:1 Decentralized Currency Mapping for ERC-20 Tokens

In this article, we show you how to map and bridge an Ethereum ERC-20 token to the Verus testnet blockchain as a currency one-to-one. We use the groundbreaking non-custodial and provable Verus-Ethereum bridge, and we don't even need to code anything!

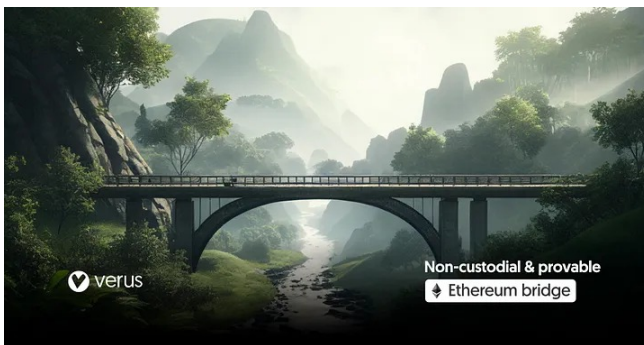


Max Theyse · Follow

Published in Verus Coin · 4 min read · Feb 9, 2023



176



Note that this walkthrough is for the Verus Testnet blockchain, which simulates

Sign up to discover human stories that deepen your understanding of the world.

Free

- ✓ Distraction-free reading. No ads.
- ✓ Organize your knowledge with lists and highlights.
- ✓ Tell your story. Find your audience.

Sign up for free

Membership

- ✓ Access the best member-only stories.
- ✓ Support independent authors.
- ✓ Listen to audio narrations.
- ✓ Read offline.
- ✓ Join the Partner Program and earn for your writing.

Try for \$5/month

- convert currencies through the bridge converter currency. Use on Ethereum as well as on Verus. [Learn more on bridge converter currencies](#)

Defining a Currency

Defining a currency on the Verus network is simple, just two commands are all you need. When the currency is successfully launched it can flow around within the protocol, and outside of the protocol (like an ERC-20, in our example). Let's get started.

VerusID Namespace

Before defining the currency, we already registered the VerusID `MaxUSDC@`. This cost us 100 VRSCTEST (or 80 when using a referral), which were distributed to the block producers. These costs can drastically vary depending on which PBaaS-chain you create a VerusID.

VerusID is the **namespace** for currencies and PBaaS-blockchains, among many other things. [Learn more about VerusID](#) @

Defining the mapped currency

Defining a currency costs 200 VRSCTEST. These costs can change depending on which blockchain you want to define a currency. When launching PBaaS-chains you can define your own costs in the chains' native currency. **Good to know:** these costs are going to the miners and stakers of the blockchain.

With a few simple commands we will:

- Define a currency mapped to the USDC Ethereum contract with a registered namespace (VerusID) on the Verus Testnet blockchain
- Export the currency to Ethereum through [the non-custodial, provable bridge](#) @

Before we can define the currency which is mapped to USDC on Ethereum, we must fund `MaxUSDC@` with enough VRSCTEST to launch the currency. And after that with enough vETH or VRSCTEST to pay for the gas fees that export the currency to Ethereum.

200+ VRSCTEST to launch the currency, and around 0.05 vETH or equivalent worth of VRSCTEST to export the currency to Ethereum.

Fund your VerusID with vETH in two ways:

1. Send ETH to Verus (into vETH) with [the Verus-Ethereum bridge](#) @
2. Convert VRSCTEST or any other currency into vETH using Verus' protocol level DeFi [more info on how to convert](#) @

The command that defines our currency `MaxUSDC`, which is mapped to the ERC-20 USDC (Goerli Testnet) [contract address on Etherscan](#) @, is as follows:

```
./verus -chain=VRSCTEST definecurrency '{
  "name": "MaxUSDC",
  "options": 32,
  "systemid": "veth",
  "parent": "vrscctest",
  "launchsystemid": "vrscctest",
  "nativecurrencyid": {
    "type": 9,
    "address": "0x98339D8C260052B7ad81c28c16C0b98420f2B46a"
  },
  "initialsupply": 0,
  "proofprotocol": 3
}'
```

★ To learn more about the many different launch options when defining a currency on Verus, [go here](#) @ We encourage anyone to start testing the protocol to get ahead of the game. [The community is happy to help on Discord](#) @

When we do the command above we get returned a HEX. Only after we make a transaction with the HEX, will the currency launch be broadcasted to

the network. See below:

```
./verus -chain=VRSCTEST sendrawtransaction "hex"
```


Now you can see the currency on the Verus network. We just need to make sure it's on Ethereum as well.

Exporting the mapped currency

Now, let's export the currency from Verus to Ethereum as an ERC-20 over the non-custodial bridge. We do this with the following command:

```
./verus chain=VRSCTEST sendcurrency "MaxUSDC@" '[[{"address": "0xAD84C9EE28FB4c3b9C6ef1B86D1ED00A82DB84E9", "currency": "MaxUSDC", "amount": 0, "exportto": "veth", "exportcurrency": true, "feecurrency": "veth"}]]'
```

Note: the first `MaxUSDC@` field is the change address for fees being returned to you, and the `address` is the Verus-Ethereum non-custodial smart contract bridge address. You can also omit `feecurrency` if you would rather pay the export fees in VRSCTEST.

After the bridge has been notarized to the blockheight where you have exported the currency, you can choose it from the token dropdown [on the bridge website](#) .


ERC-20s on Verus

After sending from Ethereum to Verus, the token is free to move through the Internet of Value on the Verus network, across PBaaS-blockchains, and also back & forth with Ethereum.

As easily as the token was first defined and launched, it can now become one of the reserves in a multi-reserve liquidity basket AMM as part of Verus DeFi.

Anybody can define a multi-reserve currency which includes `MaxUSDC` as a reserve with weighted percentage (10–90%). The reserves grow as they retain a small fee whenever assets are traded in/out/through.


A multi-reserve currency itself can also be used as one of the reserves in another multi-reserve liquidity basket AMM token on Verus.

 Stay tuned for more on this later, or start reading [docs.verus.io](#), or [join the Discord to learn](#).

. . .

Credits for the article go to community member 'ejuliano'.

. . .

 [Explore with the community on Discord](#)

[Follow on Twitter](#)

[Go to verus.io](#)

Blockchain Development

Blockchain Technology

Ethereum

Ethereum Blockchain

Cryptocurrency



Written by Max Theyse

149 Followers · Editor for Verus Coin

Follow



More from Max Theyse and Verus Coin

Max Theyse in Verus Coin

Introducing Pure—The Currency 100% Backed by Verus & Bitcoin

Pure is a decentralized currency fully backed by Verus (VRSC) & Bitcoin (tBTC).

8 min read · Mar 16, 2024

👤 196 💬

Max Theyse in Verus Coin

Verus: Profit Generating Protocol for Miners and Stakers

Future and present Verus miners: take notice. Be ready to maximize profit with your...

4 min read · May 12, 2021

👤 304 💬



Oliver in Verus Coin

How to Start CPU Mining Verus Coin VRSC from Your Computer i...

A Dummies Step by Step Guide to Pool Mining Verus Coin with a CPU!

8 min read · Jan 10, 2019

👤 346 💬 1

Max Theyse

How to preconvert into Bridge.vARRR

Participate in the first ever Verus PBaaS-chain launch—vARRR. Learn how to...

4 min read · Mar 20, 2024

👤 259 💬



See all from Max Theyse

See all from Verus Coin

Recommended from Medium

PYRIN-ONE in pyrin

PYRIN AMA (Ask me Anything)

On February 1st, 2024, the PYRIN team held its first AMA (Ask Me Anything). The team...

13 min read · Feb 2, 2024

Perzibal

How to mine TAI using TonAi on telegram ? Full guide.

Start Mining on Telegram bot for Free

2 min read · Feb 11, 2024

Lists

data science and AI
40 stories · 122 saves

Generative AI Recommended Reading
52 stories · 894 saves

Modern Marketing
103 stories · 523 saves

👤 Passive Crypto Mining

My Top 15 Passive Crypto Miners of 2024

I've spent 2 weeks compiling the most profitable Crypto Miners for 2024. You can...

11 min read · Jan 12, 2024

👤 2K 💬 29



👤 Max Theyse in Verus Coin

How-to Participate in the Verus-Ethereum Bridge Launch

Instructions on how to use the Verus-Ethereum Bridge website and Verus Deskto...

5 min read · Oct 12, 2023

👤 146 💬



👤 Albert Peter in Cryptocurrency Scripts

Top 5 Crypto Gems Predicted to Reach 100x-1000x Gains in 2024

Discover the top 5 crypto gems poised for 100x-1000x gains in 2024. Don't miss out on...

6 min read · Mar 14, 2024

👤 538 💬 13



👤 Yash Agarwal

State of Solana DePIN 2024

Dive deep into Decentralised Physical Infrastructure (DePIN) and why Solana is the...

24 min read · Feb 16, 2024

👤 150 💬



See more recommendations