Custom Gas Token Explainer

The Custom Gas Token configuration lets OP Stack chain operators deploy their chain allowing a specific ERC-20 token to be deposited in as the native token to pay for gas fees. Chain operators can now customize their gas token to:

- provide utility for their project's token
- make it easier to subsidize gas costs for their users directly via their token treasury
- facilitate in-game economies, allowing players to pay for gas with their in-game currency
- build alignment with the community of any token.

Native Gas Tokens

By default, L2 OP Stack chains allow users to deposit ETH from L1 into the L2 chain as native L2 token that can then be used to pay for gas fees. Chain operators wanted to configure the stack to use a custom token to pay for gas, other than ETH.

With custom gas tokens, chain operators can now set an L1 ERC-20 token address at the time of deploying the contracts of their L2 chain. When deposited, this L1 ERC-20 token will become the native gas token on the L2 and can be used to pay for gas fees.

⚠ Caveats chain operators should be aware of:

- You cannot switch from your custom gas token to another token after the chain is launched.
- Existing chains cannot switch to use a custom gas token. A custom token must be configured at the time of chain deployment.
- You will need to manually modify your fee configuration values to properly charge fees for users.
- Understand theconstraints required for your chain to be able to join the Superchain
- when setting the custom gas token for your chain.

How It Works

This is the general flow of how custom gas tokens work on the OP Stack:

- On deployment of an L2 chain, set the address of an L1 token to be used as the gas token.
- When depositing the custom gas token to L2, it mints the native L2 token.
- Withdrawing the native L2 token results in the unlocking of the custom token on L1.
- The native L2 token is used by default to pay for gas fees.

Benefits

Chain operators get the following benefits by using custom gas tokens:

- can use an L1 ERC-20 token as the custom gas token for L2 OP Stack chain deployment
- can use an ERC-20 token on an existing L2 OP Stack chain as as the custom gas token for an L3 OP Stack chain deployment
- can use custom gas tokens with other configurations of the OP Stack, including Plasma Mode.

Considerations

The custom token being used must adhere to the following constraints to be able to join the Superchain:

- must be a valid ERC-20 token
- the number of decimals on the token MUST be exactly 18
- the name of the token MUST be less than or equal to 32 bytes
- symbol MUST be less than or equal to 32 bytes.
- must not be yield-bearing
- cannot be rebasing or have a transfer fee
- must be transferrable only via a call to the token address itself
- must only be able to set allowance via a call to the token address itself
- must not have a callback on transfer, and more generally a user must not be able to make a transfer to themselves revert
- a user must not be able to make a transfer have unexpected side effects

Next Steps

- Ready to get started? Read our guide on how todeploy your custom gas token chain

- .
- For more info about how the custom gas token feature works under the hoodcheck out the specs(opens in a new tab)
- .

FAQs

What is the Wrapped (ERC-20) Gas Token?

- TheWETH
- predeploy at0x4200000000000000000000000000000000000006
- represents the wrapped custom gas token. If you wish to transact with your native gas token as an ERC-20, you candeposit
- andwithdraw
- from this contract to wrap and unwrap your token.
- What this means is the asset at theWETH
- predeploy is notether
- but instead is the wrapped version of your custom gas token. Its an ERC20 token, thename()
- will be"Wrapped ..."
- (whatever the name of your token is).

How do I charge fees as the chain operator?

The initial release of custom gas token does not have special logic for taking into account the exchange rate between the custom gas token and ether. Currently, the protocol charges fees for users with scalars on the L1 blob fee and L1 base fee in ETH. Chain operator will be taking user fees in the custom gas token and spending in ether.

For the initial release of custom gas token, the chain operator will need to manage this either out of band or by manually tuning the fee scalar values to capture the exchange rate between ETH and the custom gas token to ensure profitability. A future release may includeL1 Fee Abstraction(opens in a new tab) where the L1 fee is calculated in a smart contract instead of native code. This would enable an on chain oracle to take into account the exchange rate of the custom gas token.

Can I migrate my chain into being custom gas token?

If you are already live usingether to pay for gas, you cannot become a custom gas token chain. This would likely require a risky, high lift state migration that we would not recommend.

Do node operators need to do anything?

There are no core changes toop-geth orop-node so the infrastructure that node operators run is exactly the same as any other OP Stack based chain. The differences are encapsulated in the smart contracts, so only the L1 contract deployments and L2 genesis generation need to be done in a particular way.

MIPS.sol Sequencer Outages