How to use a custom gas token on your Orbit chain

PUBLIC PREVIEW, MAINNET READY Orbit chains are nowMainnet ready ! Note that Orbit is still apublic preview capability - the Orbit product and its supporting documentation may change significantly as we capture feedback from readers like you.

To provide feedback, click theRequest an update button at the top of this documentjoin the Arbitrum Discord , or reach out to our team directly by completingthis form . When deploying your AnyTrust Orbit chain you have the option of using a custom gas token, different than ETH, that is natively used for gas payments on the network. When choosing this option, there are certain requirements that the token needs to comply with, as well as certain chain configuration that needs to be adjusted. This guide covers this information.

Requirements of the custom gas token

The main requirements for a custom gas token is that it must be an ERC-20 token, and it must be natively deployed on the parent chain. During chain deployment, the gas token is "natively bridged" and then properly configured as the native gas token on the Orbit chain.

There are other important considerations to keep in mind when deciding to use a custom gas token. Restrictions on the ERC-20 token include:

- Currently, the token must be configured with 18 decimals.
- The token can't be rebasing or have a transfer fee.
- The token must only be transferrable via a call to the token address itself.
- The token must only be able to set allowance via a call to the token address itself.
- The token must not have a callback on transfer, and more generally a user must not be able to make a transfer to themselves revert.

Configuration of the Orbit chain when using a custom gas token

There are several parameter changes required to ensure proper functioning of an Orbit chain configured to use a custom gas token.

After deploying the chain, you must reset the base fees of the parent chain by calling the following functions in theArbOwner precompile:

- SetL1PricePerUnit
- , settingpricePerUnit
- to0
- SetL1PricingRewardRate
- , settingperUnitReward
- to0

Note: these methods use L1 to refer to the parent chain of the Orbit chain.

These parameter changes are strongly recommended to avoid charging users for a non-existent parent chain's base fee. The impact of not doing this is that Nitro will apply a parent chain's fee to all transactions. As Nitro assumes the native asset is ETH, all fees are expected to be denominated in ETH. This has the consequence of overcharging users for parent chain fees in native tokens that are more expensive than ETH (for example, if you use wrapped BTC as the custom gas token). In the case for tokens with prices much lower than ETH, the impact is far less pronounced.

In either case we strongly recommend taking these steps to avoid any issues with chain economics. You can read more about how Nitro manages fees inGas and fees . Edit this page Last updatedonMar 26, 2024 Previous Additional configuration parameters Next How to customize your Orbit chain's precompiles