Using Blobs

This guide walks you through how to switch to using blobs for your chain after Ecotone is activated.

This guide is intended for chains already upgraded to Ecotone. Please see theEcotone Upgrade Notice before continuing.

Switch to Using Blobs

Determine Scalar Values for Using Blobs

The first step to switching to submit data with Blobs is to calculate the scalar values you wish to set for the formula to charge users fees. To determine the scalar values to use for your chain, you can utilize thisfee parameter calculator(opens in a new tab) to get a better estimate for scalar values on your chain. Input the average transaction per day your chain is processing, the types of transactions that occur on your chain, theOP_BATCHER_MAX_CHANNEL_DURATION (opens in a new tab) you have parameterized on yourop-batcher , and the target margin you wish to charge users on top of your L1 costs. The following information is tuned to a network like OP Mainnet. For more details on fee scalar, seeTransaction Fees, Ecotone section .

Adjust Fees to Change Margins

As a chain operator, you may want to scale your scalar values up or down either because the throughput of your chain has changed and you are either filling significantly more or less of blobs, or because you wish to simply increase your margin to cover operational expenses. So, to increase or decrease your margin on L1 data costs, you would simply scale both thel1baseFeeScalar and thel1blobBaseFeeScalar by the same multiple.

For example, if you wished to increase your margin on L1 data costs by ~10%, you would do:

newBaseFeeScalar= prevBaseFeeScalar * 1.1 newBlobBaseFeeScalar = prevBlobBaseFeeScalar * 1.1

Update Your Scalar Values for Blobs

Once you have determined your idealBaseFeeScalar andBlobBaseFeeScalar , you will need to apply those values for your chain. The first step is to encode both values into a single value to be set in your L1 Config:

Theecotone-scalar utility(opens in a new tab) converts a desiredBaseFeeScalar andBlobBaseFeeScalar into a single value that can be provided to the L1SystemConfigProxy.setGasConfig method through its "scalar" argument. Here is an example of what you might see when using the utility:

. / bin / ecotone - scalar -- scalar = 1100

-- blob - scalar = 655000

base fee scalar

:

1100

blob base fee scalar

:

655000

v1 hex encoding

:

0x010000000000000000000000000000000000000000000000000009fe980000044c

uint value for the

'scalar' parameter in

SystemConfigProxy .setGasConfig ():
452312848583266388373324160190187140051835877600158453279134000734489543756 You will get a value that looks something like:0x01000000000000000000000000000000000000000000000000009fe980000044c Note that the0x01 byte at the start indicates the version of the scalar, and the hex value at the end are the encoded scalars. Depending on the tool used, you may have to input the uint representation or the hex version, but both Foundry'scast and Gnosis's Safe UI accept the hex value.

This value has to be set on Layer 1 by theowner of theSystemConfigProxy L1 Contract. Note that when callingsetGasConfig(uint256,uint256) , the first parameter is a deprecated value that can just be set to0 and the second parameter is where you will paste your encoded scalar value.

An exemplary way to create the calldata for your update transaction to the SystemConfigProxy is:

cast

calldata

'setGasConfig(uint256,uint256)'

0

0x01000000000000000000000000000000000000000000000000009fe980000044c ` Check that the gas price oracle on L2 returns the expected values forbaseFeeScalar andblobBaseFeeScalar (wait ~1 minute):

This is checked on L2, so ensure you are using an RPC URL for your chain. You'll also need to provide agas-price to geth when making this call.

baseFeeScalar blobBaseFeeScalar cast

call

0x420000000000000000000000000000000000000F

'baseFeeScalar()(uint256)'

--gas-price

10000000

--rpc-url

YOUR_L2_RPC_URL

Update Your Batcher to Post Blobs

Now that the fee config has been updated, you should immediately configure your batcher!

⚠ Your chain may be undercharging users during the time between updating the scalar values and updating the Batcher, so aim to do this immediately after. Steps to configure the batcher:

- ConfigureOP_BATCHER_DATA_AVAILABILITY_TYPE=blobs
- . The batcher will have to be restarted for it to take effect.
- Ensure yourOP_BATCHER_MAX_CHANNEL_DURATION
- is properly set to maximize your fee savings. For more info, see theChain Configuration
- guide.

Switch Back to Using Calldata

As a chain operator, if theblobBaseFee is expensive enough and your chain is not processing enough transactions to meaningfully fill blobs within your configured batcherOP_BATCHER_MAX_CHANNEL_DURATION , you may wish to switch back to posting data to calldata. Utilize thefee parameter calculator(opens in a new tab)to inform whether your transactions will be cheaper if submitting blobs or if submitting calldata. Chains can follow these steps to switch from blobs back to using calldata.

Determine Your Scalar Values for Using Calldata

If you are using calldata, then you can set yourBaseFeeScalar similarly to how you would have set "scalar" prior to Ecotone, though with a 5-10% bump to compensate for the removal of the "overhead" component. You can utilize thisfee parameter calculator(opens in a new tab) to get a better estimate for scalar values on your chain. The following information is tuned to a network like OP Mainnet.

Chains can update their fees to increase or decrease their margin. If using calldata, thenBaseFeeScalar should be scaled to achieve the desired margin. For example, to increase your L1 Fee margin by 10%:

BaseFeeScalar = BaseFeeScalar * 1.1 BlobBaseFeeScalar = 0

Update Your Scalar Values for Using Calldata

To set your scalar values, follow the same process as laid out in Update your Scalar values for Blobs .

Update Your Batcher to Post Calldata

Now that the fee config has been updated, you will want to immediately configure your batcher.

Reminder, that your chain may be undercharging users during the time between updating the scalar values and updating the Batcher, so aim to do this immediately after. * ConfigureOP_BATCHER_DATA_AVAILABILITY_TYPE=calldata * . The batcher will have to be restarted for it to take effect. * Ensure yourOP_BATCHER_MAX_CHANNEL_DURATION * is properly set to maximize savings.NOTE: * While setting a high value here will lower costs, it will be less meaningful than for low throughput chains using blobs. SeeChain Operator Configuration * for more details.

Other Considerations

- For general information on Ecotone changes, visit theEcotone notice page
- .
- If you want to enable archive nodes, you will need to access a blob archiver service. You can useOptimism's
- orrun your own
- .

Configuration Using Snap Sync