

# Batcher Configuration

This page lists all configuration options for the op-batcher. The op-batcher posts L2 sequencer data to the L1, to make it available for verifiers. The following options are from the --help in [v1.7.6\(opens in a new tab\)](#).

## Global Options

### active-sequencer-check-duration

The duration between checks to determine the active sequencer endpoint. The default value is 2m0s .

Syntax Example Environment Variable --active-sequencer-check-duration=

### approx-compr-ratio

The approximate compression ratio ( $\leq 1.0$  ). Only relevant for ratio compressor. The default value is 0.6 .

Syntax Example Environment Variable --approx-compr-ratio=

### batch-type

The batch type. 0 for SingularBatch and 1 for SpanBatch . The default value is 0 for SingularBatch .

Syntax Example Environment Variable --batch-type=

### check-recent-txs-depth

Indicates how many blocks back the batcher should look during startup for a recent batch tx on L1. This can speed up waiting for node sync. It should be set to the verifier confirmation depth of the sequencer (e.g. 4). The default value is 0 .

Syntax Example Environment Variable --check-recent-txs-depth=

### compression-algo

The compression algorithm to use. Valid options: zlib, brotli, brotli-9, brotli-10, brotli-11. The default value is zlib .

Syntax Example Environment Variable --compression-algo=

### compressor

The type of compressor. Valid options: none, ratio, shadow. The default value is shadow .

Syntax Example Environment Variable --compressor=

### data-availability-type

The data availability type to use for submitting batches to the L1. Valid options: calldata, blobs. The default value is calldata .

Syntax Example Environment Variable --data-availability-type=

### fee-limit-multiplier

The multiplier applied to fee suggestions to put a hard limit on fee increases. The default value is 5 .

Syntax Example Environment Variable --fee-limit-multiplier=

### hd-path

The HD path used to derive the sequencer wallet from the mnemonic. The mnemonic flag must also be set.

Syntax Example Environment Variable --hd-path=

### l1-eth-rpc

HTTP provider URL for L1.

Syntax Example Environment Variable --l1-eth-rpc=

## **l2-eth-rpc**

HTTP provider URL for L2 execution engine. A comma-separated list enables the active L2 endpoint provider. Such a list needs to match the number of rollup-rpcs provided.

Syntax Example Environment Variable --l2-eth-rpc=

## **log.color**

Color the log output if in terminal mode. The default value is false .

Syntax Example Environment Variable --log.color=

## **log.format**

Format the log output. Supported formats: 'text', 'terminal', 'logfmt', 'json', 'json-pretty'. The default value is text .

Syntax Example Environment Variable --log.format=

## **log.level**

The lowest log level that will be output. The default value is INFO .

Syntax Example Environment Variable --log.level=

## **max-channel-duration**

The maximum duration of L1-blocks to keep a channel open. 0 to disable. The default value is 0 .

Syntax Example Environment Variable --max-channel-duration=

## **max-l1-tx-size-bytes**

The maximum size of a batch tx submitted to L1. Ignored for blobs, where max blob size will be used. The default value is 120000 .

Syntax Example Environment Variable --max-l1-tx-size-bytes=

## **max-pending-tx**

The maximum number of pending transactions. 0 for no limit. The default value is 1 .

Syntax Example Environment Variable --max-pending-tx=

## **metrics.addr**

Metrics listening address. The default value is 0.0.0.0 .

Syntax Example Environment Variable --metrics.addr=

## **metrics.enabled**

Enable the metrics server. The default value is false .

Syntax Example Environment Variable --metrics.enabled=

## **metrics.port**

Metrics listening port. The default value is 7300 .

Syntax Example Environment Variable --metrics.port=

## **mnemonic**

The mnemonic used to derive the wallets for either the service.

Syntax Example Environment Variable --mnemonic=

## **network-timeout**

Timeout for all network operations. The default value is 10s .

Syntax Example Environment Variable --network-timeout=

### **num-confirmations**

Number of confirmations which we will wait after sending a transaction. The default value is 10 .

Syntax Example Environment Variable --num-confirmations=

### **plasma.da-server**

HTTP address of a DA Server.

Syntax Example Environment Variable --plasma.da-server=

### **plasma.da-service**

Use DA service type where commitments are generated by plasma server. The default value is false .

Syntax Example Environment Variable --plasma.da-service=

### **plasma.enabled**

Enable plasma mode. The default value is false .

Syntax Example Environment Variable --plasma.enabled=

### **plasma.verify-on-read**

Verify input data matches the commitments from the DA storage service. The default value is true .

Syntax Example Environment Variable --plasma.verify-on-read=

### **poll-interval**

How frequently to poll L2 for new blocks. The default value is 6s .

Syntax Example Environment Variable --poll-interval=

### **pprof.addr**

pprof listening address. The default value is 0.0.0.0 .

Syntax Example Environment Variable --pprof.addr=

### **pprof.enabled**

Enable the pprof server. The default value is false .

Syntax Example Environment Variable --pprof.enabled=

### **pprof.path**

pprof file path. If it is a directory, the path is {dir}/{profileType}.prof .

Syntax Example Environment Variable --pprof.path=

### **pprof.port**

pprof listening port. The default value is 6060 .

Syntax Example Environment Variable --pprof.port=

### **pprof.type**

pprof profile type. One of cpu, heap, goroutine, threadcreate, block, mutex, allocs.

Syntax Example Environment Variable --pprof.type=

## **private-key**

The private key to use with the service. Must not be used with mnemonic.

Syntax Example Environment Variable --private-key=

## **resubmission-timeout**

Duration we will wait before resubmitting a transaction to L1. The default value is 48s .

Syntax Example Environment Variable --resubmission-timeout=

## **rollup-rpc**

HTTP provider URL for Rollup node. A comma-separated list enables the active L2 endpoint provider. Such a list needs to match the number of l2-eth-rpcs provided.

Syntax Example Environment Variable --rollup-rpc=

## **rpc.addr**

rpc listening address. The default value is 0.0.0.0 .

Syntax Example Environment Variable --rpc.addr=

## **rpc.enable-admin**

Enable the admin API. The default value is false .

Syntax Example Environment Variable --rpc.enable-admin=

## **rpc.port**

rpc listening port. The default value is 8545 .

Syntax Example Environment Variable --rpc.port=

## **safe-abort-nonce-too-low-count**

Number of ErrNonceTooLow observations required to give up on a tx at a particular nonce without receiving confirmation. The default value is 3 .

Syntax Example Environment Variable --safe-abort-nonce-too-low-count=

## **sequencer-hd-path**

DEPRECATED: The HD path used to derive the sequencer wallet from the mnemonic. The mnemonic flag must also be set.

Syntax Example Environment Variable --sequencer-hd-path=

## **signer.address**

Address the signer is signing transactions for.

Syntax Example Environment Variable --signer.address=

## **signer.endpoint**

Signer endpoint the client will connect to.

Syntax Example Environment Variable --signer.endpoint=

## **signer.tls.ca**

tls ca cert path. The default value is tls/ca.crt .

Syntax Example Environment Variable --signer.tls.ca=

## **signer.tls.cert**

tls cert path. The default value istls/tls.crt .

Syntax Example Environment Variable --signer.tls.cert=

### **signer.tls.key**

tls key. The default value istls/tls.key .

Syntax Example Environment Variable --signer.tls.key=

### **stopped**

Initialize the batcher in a stopped state. The batcher can be started using the admin\_startBatcher RPC. The default value isfalse .

Syntax Example Environment Variable --stopped=

### **sub-safety-margin**

The batcher tx submission safety margin (in #L1-blocks) to subtract from a channel's timeout and sequencing window, to guarantee safe inclusion of a channel on L1. The default value is10 .

Syntax Example Environment Variable --sub-safety-margin=

### **target-num-frames**

The target number of frames to create per channel. Controls number of blobs per blob tx, if using Blob DA. The default value is1 .

Syntax Example Environment Variable --target-num-frames=

### **txmgr.fee-limit-threshold**

The minimum threshold (in GWei) at which fee bumping starts to be capped. Allows arbitrary fee bumps below this threshold. The default value is100 .

Syntax Example Environment Variable --txmgr.fee-limit-threshold=

### **txmgr.min-basefee**

Enforces a minimum base fee (in GWei) to assume when determining tx fees. 1 GWei by default. The default value is1 .

Syntax Example Environment Variable --txmgr.min-basefee=

### **txmgr.min-tip-cap**

Enforces a minimum tip cap (in GWei) to use when determining tx fees. 1 GWei by default. The default value is1 .

Syntax Example Environment Variable --txmgr.min-tip-cap=

### **txmgr.not-in-mempool-timeout**

Timeout for aborting a tx send if the tx does not make it to the mempool. The default value is2m0s .

Syntax Example Environment Variable --txmgr.not-in-mempool-timeout=

### **txmgr.receipt-query-interval**

Frequency to poll for receipts. The default value is12s .

Syntax Example Environment Variable --txmgr.receipt-query-interval=

### **txmgr.send-timeout**

Timeout for sending transactions. If 0 it is disabled. The default value is0s .

Syntax Example Environment Variable --txmgr.send-timeout=

### **wait-node-sync**

Indicates if, during startup, the batcher should wait for a recent batcher tx on L1 to finalize (via more block confirmations). This should help avoid duplicate batcher txs. The default value is false .

Syntax Example Environment Variable --wait-node-sync=

## Miscellaneous

### help

Show help. The default value is false.

Syntax Example Environment Variable --help=

### version

Print the version. The default value is false.

Syntax Example Environment Variable --version=

## Recommendations

### Set Your `OP_BATCHER_MAX_CHANNEL_DURATION`

The default value inside `op-batcher` , if not specified, is still 0 , which means channel duration tracking is disabled. For very low throughput chains, this would mean to fill channels until close to the sequencing window and post the channel to L1 `SUB_SAFETY_MARGIN` L1 blocks before the sequencing window expires. To minimize costs, we recommend setting your `OP_BATCHER_MAX_CHANNEL_DURATION` to target 5 hours, with a value of 1500 L1 blocks. When non-zero, this parameter is the max time (in L1 blocks, which are 12 seconds each) between which batches will be submitted to the L1. If you have this set to 5 for example, then your batcher will send a batch to the L1 every  $5 \times 12 = 60$  seconds. When using blobs, because 130kb blobs need to be purchased in full, if your chain doesn't generate at least ~130kb of data in those 60 seconds, then you'll be posting only partially full blobs and wasting storage.

- We do not recommend setting any values higher than targeting 5 hours, as batches have to be submitted within the sequencing window which defaults to 12 hours for OP chains, otherwise your chain may experience a 12 hour long chain reorg. 5 hours is the longest length of time we recommend that still sits snugly within that 12 hour window to avoid affecting stability.
- If your chain fills up full blobs of data before the `OP_BATCHER_MAX_CHANNEL_DURATION` elapses, a batch will be submitted anyways - (e.g. even if the OP Mainnet batcher sets an `OP_BATCHER_MAX_CHANNEL_DURATION` of 5 hours, it will still be submitting batches every few minutes)

⚠ While setting an `OP_BATCHER_MAX_CHANNEL_DURATION` of 1500 results in the cheapest fees, it also means that your [safe head \(opens in a new tab\)](#) can stall for up to 5 hours.

- This will negatively impact apps on your chain that rely on the safe head for operation. While many apps can likely operate simply by following the unsafe head, often Centralized Exchanges or third party bridges wait until transactions are marked safe before processing deposits and withdrawal.
- Thus a larger gap between posting batches can result in significant delays in the operation of certain types of high-security applications.

### Configure Your Batcher to Use Multiple Blobs

The `op-batcher` has the capabilities to send multiple blobs per single blob transaction. This is accomplished by the use of multi-frame channels, see the [specs \(opens in a new tab\)](#) for more technical details on channels and frames.

A minimal batcher configuration (with env vars) to enable 6-blob batcher transactions is:

- `OP_BATCHER_BATCH_TYPE=1` # span batches, optional
- `OP_BATCHER_DATA_AVAILABILITY_TYPE=blobs`
- `OP_BATCHER_TARGET_NUM_FRAMES=6` # 6 blobs per tx
- `OP_BATCHER_TXMGR_MIN_BASEFEE=2.0` # 2 gwei, might need to tweak, depending on gas market
- `OP_BATCHER_TXMGR_MIN_TIP_CAP=2.0` # 2 gwei, might need to tweak, depending on gas market
- `OP_BATCHER_RESUBMISSION_TIMEOUT=240s` # wait 4 min before bumping fees This enables blob transactions and sets the target number of frames to 6, which translates to 6 blobs per transaction. The minimum tip cap and base fee are also lifted to 2 gwei because it is uncertain how easy it will be to get 6-blob transactions included and slightly higher priority fees should help. The resubmission timeout is increased to a few minutes to give more time for inclusion before bumping the fees because current transaction pool implementations require a doubling of fees for blob transaction replacements.

Multi-blob transactions are particularly useful for medium to high-throughput chains, where enough transaction volume exists to fill up 6 blobs in a reasonable amount of time. You can use [this calculator \(opens in a new tab\)](#) for your chain to determine what number of blobs are right for you, and what gas scalar configuration to use. Please also refer to guide on [Using Blobs](#) for chain operators.

[Rollup Configuration](#) [Proposer Configuration](#)