Chain Configuration

The OP Stack is a flexible platform with various configuration values that you can tweak to fit your specific needs. If you're looking to fine-tune your OP Stack chain deployment, look no further.

⚠ Work in Progress

OP Stack configuration is an active work in progress and will likely evolve significantly as time goes on. If something isn't working about your configuration, check back with this page to see if anything has changed.

New Blockchain Configuration

New OP Stack blockchains are currently configured with a JSON file inside the Optimism repository. The file is/packages/contracts-bedrock/deploy-config/.json . For example,[this is the configuration file for the tutorial blockchain(opens in a new tab)](#) .

Admin addresses

Key Type Description Default / Recommended value finalSystemOwner L1 Address Address that will own all ownable contracts on L1 once the deployment is finished, including theProxyAdmin contract. It is recommended to have a single admin address to retain a common security model. proxyAdminOwner L2 Address Address that will own theProxyAdmin contract on L2. The L2ProxyAdmin contract owns all of theProxy contracts for every predeployed contract in the range0x42...0000 to0x42..2048 . This makes predeployed contracts easily upgradeable. It is recommended to have a single admin address to retain a common security model.

Fee recipients

Key Type Description Default value baseFeeVaultRecipient L1 or L2 Address Address that the base fees from all transactions on the L2 can be withdrawn to. It is recommended to have a single admin address to retain a common security model. l1FeeVaultRecipient L1 or L2 Address Address that the L1 data fees from all transactions on the L2 can be withdrawn to. It is recommended to have a single admin address to retain a common security model. sequencerFeeVaultRecipient L1 or L2 Address Address that the tip fees from all transactions on the L2 can be withdrawn to. It is recommended to have a single admin address to retain a common security model.

Minimum Fee Withdrawal Amounts

Key Type Description Default value baseFeeVaultMinimumWithdrawalAmount Number in wei The minimum amount of ETH theBaseFeeVault contract must have for a fee withdrawal. 10 ether l1FeeVaultMinimumWithdrawalAmount Number in wei The minimum amount of ETH theL1FeeVault contract must have for a fee withdrawal. 10 ether sequencerFeeVaultWithdrawalAmount Number in wei The minimum amount of ETH theSequencerFeeVault contract must have for a fee withdrawal. 10 ether

Withdrawal Network

Key Type Description Default value baseFeeVaultWithdrawalNetwork Number representing network enum A value of0 will withdraw ETH to the recipient address on L1 and a value of1 will withdraw ETH to the recipient address on L2. l1FeeVaultWithdrawalNetwork Number representing network enum A value of0 will withdraw ETH to the recipient address on L1 and a value of1 will withdraw ETH to the recipient address on L2. sequencerFeeVaultWithdrawalNetwork Number representing network enum A value of0 will withdraw ETH to the recipient address on L1 and a value of1 will withdraw ETH to the recipient address on L2.

Misc.

Key Type Description Default value numDeployConfirmations Number of blocks Number of confirmations to wait when deploying smart contracts to L1. 1 l1StartingBlockTag Block hash Block tag for the L1 block where the L2 chain will begin syncing from. Generally recommended to use a finalized block to avoid issues with reorgs. l1ChainID Number Chain ID of the L1 chain. 1 for L1 Ethereum mainnet, 11155111 for the Sepolia test network.[See here for other blockchains(opens in a new tab)](#) . l2ChainID Number Chain ID of the L2 chain. 42069

Blocks

These fields apply to L2 blocks: Their timing, when do they need to be written to L1, and how they get written.

Key Type Description Default value l2BlockTime Number of seconds Number of seconds between each L2 block. Must be < = L1 block time (12 on mainnet and Sepolia) 2 maxSequencerDrift Number of seconds How far the L2 timestamp can differ from the actual L1 timestamp 600 (10 minutes) sequencerWindowSize Number of blocks Maximum number of L1 blocks that a Sequencer can wait to incorporate the information in a specific L1 block. For example, if the window is10 then the information in L1 blockn must be incorporated by L1 blockn+10 . 3600 (12 hours) channelTimeout Number of blocks Maximum number of L1 blocks that a transaction channel frame can be considered valid. A transaction channel frame is a chunk of a compressed batch of transactions. After the timeout, the frame is dropped. 300 (1 hour) p2pSequencerAddress L1 Address Address of the key that the Sequencer uses to sign blocks on the p2p network. Sequencer, an address for which you own the private key batchInboxAddress L1 Address Address that Sequencer transaction batches are sent to on L1. 0xff00…0042069 batchSenderAddress L1 Address Address that nodes will filter for when searching for Sequencer transaction batches being sent to thebatchInboxAddress . Can be updated later via theSystemConfig contract on L1. Batcher, an address for which you own the private key

Proposal Fields

These fields apply to output root proposals. Thel2OutputOracleSubmissionInterval is configurable, see the section below for guidance.

Key Type Description Default value l2OutputOracleStartingBlockNumber Number Block number of the first OP Stack block. Typically this should be zero, but this may be non-zero for networks that have been upgraded from a legacy system (like OP Mainnet). Will be removed with the addition of permissionless proposals. 0 l2OutputOracleStartingTimestamp Number Timestamp of the first OP Stack block. This MUST be the timestamp corresponding to the block defined by thel1StartingBlockTag . Will be removed with the addition of permissionless proposals. l2OutputOracleSubmissionInterval Number of blocks Number of blocks between proposals to theL2OutputOracle . Will be removed with the addition of permissionless proposals. 120 (4 minutes) finalizationPeriodSeconds Number of seconds Number of seconds that a proposal must be available to challenge before it is considered finalized by theOptimismPortal contract. Recommend 12 on test networks, seven days on production ones l2OutputOracleProposer L1 Address Address that is allowed to submit output proposals to theL2OutputOracle contract. Will be removed when the OP Stack has permissionless proposals. l2OutputOracleChallenger L1 Address Address that is allowed to challenge output proposals submitted to theL2OutputOracle . Will be removed when the OP Stack has permissionless challenges. It is recommended to have a single admin address to retain a common security model.

Setting yourl2OutputOracleSubmissionInterval

When deploying your contracts, you can set thel2OutputOracleSubmissionInterval to any value you wish to save on costs (e.g., 24 hours or 12 hours). On OP mainnet,l2OutputOracleSubmissionInterval is set to1800 L2 blocks = 1 hr . The tradeoff here is that users will only be able to prove withdrawals after an output root is posted, so users will have to submit their L2 withdrawal transaction, wait up to the number of hours in the submission interval, then prove on L1, then wait 1 week, then finalize.

After fault proofs are shipped to Mainnet, output roots can be submitted as infrequently as you wish, and there won't be a submission interval anymore.

L1 data fee

Bedrock & Regolith

These fields apply to the cost of theL1 data fee for L2 transactions prior to the Ecotone upgrade.

Key Type Description Default value gasPriceOracleOverhead Number Fixed L1 gas overhead per transaction. 2100 gasPriceOracleScalar Number Dynamic L1 gas overhead per transaction, given in 6 decimals. Default value of 1000000 implies a dynamic gas overhead of exactly 1x (no overhead). 1000000

Ecotone

For more information, see theEcotone Upgrade guide for chain operators.

Key Type Description Default value gasPriceOracleBaseFeeScalar Number Scalar applied to the Ethereum base fee in the L1 data fee cost function, given in 6 decimals. 1000000 gasPriceOracleBlobBaseFeeScalar Number Scalar applied to the Ethereum blob base fee in the L1 data fee cost function, given in 6 decimals. 0

EIP 1559 gas algorithm

These fields apply tothe EIP 1559 algorithm(opens in a new tab) used for theL2 execution costs of transactions on the blockchain.

Key Type Description Default value Value on L1 Ethereum eip1559Denominator Number Denominator used for theEIP1559 gas pricing mechanism on L2(opens in a new tab) . A larger denominator decreases the amount by which the base fee can change in a single block. 50 8 eip1559Elasticity Number Elasticity for theEIP1559 gas pricing mechanism on L2(opens in a new tab) . A larger elasticity increases the maximum allowable gas limit per block. 10 2 l2GenesisBlockGasLimit String Initial block gas limit, represented as a hex string. Default is 25m, implying a 2.5m target when combined with a 10x elasticity. 0x17D7840 l2GenesisBlockBaseFeePerGas String Initial base fee, used to avoid an unstable EIP1559 calculation out of the gate. Initial value is 1 gwei. 0x3b9aca00

Governance token

The governance token is a side-effect of use of the OP Stack in the OP Mainnet network. It may not be included by default in future releases.

Key Type Description Default value governanceTokenOwner L2 Address Address that will own the token contract deployed by default to every OP Stack based chain. governanceTokenSymbol String Symbol for the token deployed by default to each OP Stack chain. OP governanceTokenName String Name for the token deployed by default to each OP Stack chain. Optimism

OP-Batcher Configuration

TheOP_BATCHER_MAX_CHANNEL_DURATION is configurable, see the section below for guidance.

Key Type Description Default value OP_BATCHER_MAX_CHANNEL_DURATION Number of L1 Blocks The max time (in L1 blocks, which are 12 seconds each) between which batches will be submitted to the L1. 1500 The default value insideop-batcher , if not specified, is still0 , 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 toL1 SUB_SAFETY_MARGIN L1 blocks before the sequencing window expires.

Setting yourOP_BATCHER_MAX_CHANNEL_DURATION

To minimize costs, we recommend setting yourOP_BATCHER_MAX_CHANNEL_DURATION to target 5 hours, with a value of1500 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*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 theOP_BATCHER_MAX_CHANNEL_DURATION
- elapses, a batch will be submitted anyways - (e.g. even if the OP Mainnet batcher sets anOP_BATCHER_MAX_CHANNEL_DURATION
- of 5 hours, it will still be submitting batches every few minutes)

⚠ While setting anOP_BATCHER_MAX_CHANNEL_DURATION of1500 results in the cheapest fees, it also means that yoursafe 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.

Modifying Predeployed Contracts Using Blobs