

Preparing for Fjord Breaking Changes

This page outlines breaking changes related to the Fjord network upgrade for wallets and front-end developers, chain operators, and node operators. If you experience difficulty at any stage of this process, please reach out to [developer support\(opens in a new tab\)](#) .

The Fjord upgrade for OP Sepolia was activated on 1716998400 Wed May 29 16:00:00 UTC 2024 . The Fjord OP Mainnet upgrade will be optimistically activated 1720627201 Thu July 10 16:00:01 UTC 2024 , pending [governance approval\(opens in a new tab\)](#) .

What's Included in Fjord

The [Fjord network upgrade\(opens in a new tab\)](#) includes the following:

- [RIP-7212\(opens in a new tab\)](#)
- : Precompile for secp256r1 Curve Support, to reduce gas costs of many smart wallet applications.
- [Brotli\(opens in a new tab\)](#)
- as a channel compression option, for ~5-15% lower data availability costs.
- Parameter changes: * [Max sequencer drift\(opens in a new tab\)](#)
- - becomes a constant with value increased to 1800 seconds
- - 10x increased values for MAX_RLP_BYTES_PER_CHANNEL
- - and MAX_CHANNEL_BANK_SIZE
- - ([spec\(opens in a new tab\)](#)
- -)
- The [Fjord hardfork activation block\(opens in a new tab\)](#)
- includes several transactions to perform all L2 contract deployments, upgrades, enablements, and proxy updates.
- L1 Gas Cost changes: * [FastLZ\(opens in a new tab\)](#)
- - based L1 fee [cost calculation\(opens in a new tab\)](#)
- - with an upgraded [GasPriceOracle L2 predeploy\(opens in a new tab\)](#)
- - to compute it
- - GasPriceOracle
- - gets a new function `getL1FeeUpperBound`
- - as a cheap new way to calculate an upper bound for the max fee of a new transaction
- - `getL1GasUsed`
- - method of the `GasPriceOracle`
- - contract ([spec\(opens in a new tab\)](#)
- -) is being deprecated
- - `L1GasUsed`
- - field of the transaction receipt ([spec\(opens in a new tab\)](#)
- -) is being deprecated

For Wallets and Front-End Developers

The proposed Fjord upgrade to the OP Stack and OP Mainnet changes the formula for estimating the [L1 Data Fee](#) component of the [OP Stack Transaction Fee](#) .

- `getL1Fee`
- on the [GasPriceOracle contract\(opens in a new tab\)](#)
- has been updated. It now performs FastLZ compression on-chain, which is a better approximation of the

compressibility of a transaction. Combined with a linear regression model, this gives a more accurate prediction of L1 data fees.

- `getL1GasUsed`
- and the corresponding `L1GasUsed`
- field of transaction receipts are being deprecated as they no longer accurately reflect gas usage as of Ecotone. The function and field will remain; however, their usefulness is limited as they still assume calldata batching.
- `getL1Fee`
- should be used when trying to predict L1 Data fees.
- `getL1FeeUpperBound`
- is a new method to estimate fee upper bounds when sending transactions. It is much cheaper, in gas costs, than previous methods. This is what wallets and front-ends should use in practice in most cases.
- Read the [Fjord Formula section](#)
- of the [Transaction Fees](#)
- page for more information about the new formula.

Your application may need to be updated to account for this change. Read below to learn how specific changes in the Fjord upgrade require updates to your application.

Preparing Your Wallet or Front-End

Changes to the L1 Data Fee formula may affect your application if you are computing this fee component on your own. It's strongly recommended that you use [existing tooling](#) to estimate transaction fees instead of computing them yourself.

- If you cannot use existing tooling, use the `getL1Fee`
- function on the `GasPriceOracle`
- smart contract to compute the L1 Data Fee component of the transaction fee. Avoid implementing the formula yourself, as it may change in the future.
- Alternatively, you should consider using `getL1FeeUpperBound`
- if you only need to estimate an upper bound of the L1 fee for the purpose of transaction sending.

For Chain Operators

The proposed Fjord upgrade impacts OP chains and requires chain operators to upgrade their chain and configure the sequencer for Fjord.

- [Max sequencer drift \(opens in a new tab\)](#)
- becomes a constant with value increased to 1800 seconds. This gives chain operators more time to respond to L1 node issues without facing a potential L2 chain halt.
- [Brotli \(opens in a new tab\)](#)
- is now supported as a channel compression option, for ~5-15% lower data availability costs.
- An update of the fee scalars on the `SystemConfig`
- is necessary, similar to Ecotone.

PrepareSystemConfig

Transaction

An onchain transaction will be required to update the scalar for Fjord. This needs to be prepared days in advance before the activation to ensure chain operators don't operate at a loss when Fjord activates.

- Encode the scalar value using the [ecotone scalar encoding tool \(opens in a new tab\)](#)
- Send `asetGasConfig`
- transaction to `SystemConfig`
- `SetBaseFeeScalar`
- and `BlobBaseFeeScalar`
- values based on the [Fjord calculator \(opens in a new tab\)](#)

Prepare Sequencer Node

⚠ If you are operating an OP Chain that has an entry in the [superchain-registry \(opens in a new tab\)](#), the Fjord activation date is part of the op-node and op-geth nodes, and are using the `--network` and `--op-network` flags. No action is needed for the sequencer after preparing the `SystemConfig` transaction. Please skip to [Step 3: Prepare Batchers](#). For custom chains not included in the [superchain-registry \(opens in a new tab\)](#), you will need to manually configure the [activation timestamp \(opens in a new tab\)](#). You have two configuration options for your sequencer node:

- Option 1:
- Set the Fjord activation date in your `rollup.json`
- config file. You will still need to set the `override.fjord`
- flag in op-geth
- with the UNIX timestamp.

- Option 2:
- Alternatively, chain operators can use the override flags to configure your sequencer node by specifying a time in the future when Fjord will activate.* Setoverride.fjord
- - in bothop-node
- - andop-geth
- - to the UNIX timestamp of the block you want to activate the Fjord hardfork or corresponding env vars for this.
- - In general, runop-node --help
- - orop-geth --help
- - to see flags, their descriptions and environment variables.

Prepare Batcher

Preparing your batcher to activate Brotli compression is optional but recommended to achieve better channel compression.

- You can activate Brotli compression for your batcher by setting thecompression-algo
- flag.* brotli-10
- - is the recommended Brotli level and works fine for most chain configurations.
- - However, chain operators can experiment withbrotli-11
- - if it gives them better compression and their batcher can still keep up with the increased compression computation needs.

brotli defaults tobrotli-10 . If the flag is unset, it still defaults tozlib . * You can also run the batcher help to see available options:go run ./op-batcher/cmd --help |less

-- compression - algo value (default: zlib) (OP_BATCHER_COMPRESSION_ALGO) The compression algorithm to use . Valid options : zlib , brotli , brotli - 9 , brotli - 10 , brotli - 11 To verify proper configuration, chain operators should confirm in the startup logs of theirop-node andop-geth that the correct Fjord activation timestamps are set.

For Node Operators

Node operators will need to upgrade to Fjord before the activation date. For Sepolia, the op-node release[v1.7.7\(opens in a new tab\)](#) and op-geth release[v1.101315.2\(opens in a new tab\)](#) contain these changes.

These following steps are necessary for EVERY node operator:

Update to the Latest Release

- [op-geth \(opens in a new tab\)](#)
- [op-node \(opens in a new tab\)](#)

Configure the Fjord Activation Date

⚠ If you are operating a node for an OP Chain that has an entry in the[superchain-registry \(opens in a new tab\)](#) , the Fjord activation date is part of theop-node andop-geth nodes. So, no action is needed for the sequencer after upgrading to the latest release. Please skip to[Step 3: Verify Your Configuration](#) . For node operators of custom chains not included in the[superchain-registry \(opens in a new tab\)](#) , you will need to manually configure the[activation timestamp\(opens in a new tab\)](#) . This can be done one of two ways:

- Option 1:
- Set the activation time in therollup.json
- forop-node
- . You will still need to set theoverride.fjord
- flag inop-geth
- if you use this option.
- Option 2:
- Set the activation time via overrides (CLI) in bothop-node
- andop-geth
- . These will need to be set onop-node
- andop-geth

- for the sequencer and all other nodes.

Verify Your Configuration

Make the following checks to verify that your node is properly configured.

- op-node
- andop-geth
- will log their configurations at startup
- Check that the Fjord time is set to activation-timestamp
- in the op-node startup logs
- Check that the Fjord time is set to activation-timestamp
- in the op-geth startup logs

[Preparing for Fault Proofs Breaking Changes Overview](#)