How to configure your Orbit chain's node using the Orbit SDK

This guide will walk you through configuring an Orbit node supporting aRollup orAnyTrust chain.

UNDER CONSTRUCTION This document is under construction and may change significantly as we incorporatestyle guidance and feedback from readers. Feel free to request specific clarifications by clicking theRequest an update button at the top of this document. info See the"prepare-node-config" example in the Orbit SDK repository for additional guidance. * Prerequisite: having deployed an Orbit chain. If you haven't done so yet, you can find detailed instructions in theRollup Deployment Parameters * section of the rollup deployment guide.

Once you have successfully deployed and initialized the Orbit core contracts, the next step is configuring and running your Orbit chain using a Node ConfigJSON file describing all the configurations for the Arbitrum Node, including settings for the Batch-poster, Validator, and the chain itself.

Example for a Rollup Orbit Chain:

```
{ 'chain' :

{ 'info-json' :

stringifyInfoJson ( [ ... ] ) , 'name' : chainName , // Additional chain-specific configurations } , 'parent-chain' :

{ connection :

{ url : parentChainRpcUrl , } , } , 'http' :

{ addr :

'0.0.0.0' , port :

8449 , vhosts :

'*' , corsdomain :

'*' , api :

[ 'eth' ,

'net' ,

'web3' ,

'arb' ,

'debug' ] , } , 'node' :

{ // Node-specific settings including sequencer, batch-poster, staker configurations } , } ;
```

Here are some inputs details from the example above:

Parameters Description chain Details about the hosted chain, including chain ID, name, and core contracts. parent-chain Information for connecting to the parent chain. http Configuration parameters fot the HTTP server. node Specific node settings, including sequencer and batch-poster configurations.

Additional Configuration for AnyTrust Orbit Chains:

For AnyTrust Orbit chains, the Node ConfigJSON has an additional segment under thenode field. This addition includes settings specific to the AnyTrust model, such as:

- Sequencer's inbox address
- Parent chain node URL
- RPC aggregators

Example addition for AnyTrust Node Config:

```
{ ... 'node' :

{ ... 'sequencer-inbox-address' : coreContracts . sequencerInbox , 'parent-chain-node-url' : parentChainRpcUrl , 'rest-aggregator' :
```

```
{ enable :

true , urls :

'http://localhost:9876' , } , 'rpc-aggregator' :

{ 'enable' :

true , 'assumed-honest' :

1 , 'backends' :

stringifyBackendsJson ( [ ... ] ) , } , } ... } ;
```

Preparing Your Node Config File

The Node Config file includes three types of fields:

1. Information from the Orbit Deployment Chain
2. : Such as the addresses of the core contracts.
3. Parameters Configurable by the Chain Deployer
4. : These parameters, likemax-block-speed
5. , can be adjusted to modify your chain's behavior.
6. Fields Not Typically Configured
7. : Like the HTTP section, which usually remains standard.

Let's explore the parameters allowing you to set up a stable, and secure Orbit chain tailored to your project's requirements:

Node Config Generation With Orbit SDK

Generating a Node ConfigJSON file to initiate your Orbit chain is a step in the deployment process. The Orbit SDK simplifies this task with an API namedprepareNodeConfig . This API takes specific parameters for your Orbit chain and returns aJSON file that can be used as the Node Config to initiate the chain.

Here’s an example of using theprepareNodeConfig API to generate the node config:

```
const nodeConfig =

prepareNodeConfig ( { chainName :

'My Orbit Chain' , chainConfig , coreContracts , batchPosterPrivateKey :

'BATCH_POSTER_PRIVATE_KEY_HERE' , validatorPrivateKey :

'VALIDATOR_PRIVATE_KEY_HERE' , parentChainId : parentChain_chain_id , parentChainRpcUrl :
parentChain_RPC_URL , } ) ;
```

Here are some details about the parameters used in the example above:

Parameters Description chainName The name you have chosen for your Orbit chain. chainConfig Configuration used forchain deployment , returned from thecreateRollupPrepareTransactionReceipt API. coreContracts Addresses of your newly deployed Orbit chain's, also returned from thecreateRollupPrepareTransactionReceipt API. batchPosterPrivateKey Private key of the batch-poster account, used for signing batch-posting transactions and related functions. validatorPrivateKey Private key of the validator(s), used for validating state, posting Rollup Blocks (RBlocks) to the parent chain, and initiating challenges if necessary. parentChainId Chain ID of the parent chain where your Orbit chain is deployed. parentChainRpcUrl Parent chain's RPC URL. In case you do not have thechainConfig andcoreContracts readily available, you can obtain them using thecreateRollupPrepareTransaction andcreateRollupPrepareTransactionReceipt APIs.

Here's an example of how to extractchainConfig andcoreContracts using just the transaction hash from your deployment:

```
import

{ ChainConfig , createRollupPrepareTransaction , createRollupPrepareTransactionReceipt }

from

'@arbitrum/orbit-sdk' ;

const tx =

createRollupPrepareTransaction ( {

hash : txHash } ) ; const txReceipt =
createRollupPrepareTransactionReceipt ( {

hash : txHash } ) ; const

chainConfig :

ChainConfig

=

JSON . parse ( tx . getInputs ( ) [ 0 ] . config . chainConfig ) ; const coreContracts = txReceipt . getCoreContracts ( ) ;
```

This process ensures that all necessary configurations and contract details are included in your Node Config, paving the way for a successful initiation and operation of your Orbit chain.

Edit this page Last updatedonApr 2, 2024 Previous Deploy a custom gas token chain Next Deploy a token bridge