How to set up an AEP fee router

Quick start

- You can adopt the AEP Fee Router by using theAEP Router deployment scripts
- provided in theOrbit SDK

Canonical Contracts

Network Contract Address Configured For Ethereum Parent2ChildRouter https://etherscan.io/address/0x40Cd7D713D7ae463f95cE5d342Ea6E7F5cF7C999 ETH ,ERC-20 Arbitrum Nova Child2ParentRouter https://nova.arbiscan.io/address/0x36D0170D92F66e8949eB276C3AC4FEA64f83704d ETH

The AEP fee router contract system

Link to theRouter contracts' source code .

RewardDistributor

TheAEP fee router system relies on configuring an escrow contract as the intended reward address for protocol fee components. This intermediary contract is known as theRewardDistributor.

TheRewardDistributor is configured to separate the AEP portion of the fees from fees intended for the chain owner. TheRewardDistributor can be permissionlessly called to perform a withdrawal which simultaneously transfers 90% of accrued fees to the chain's fee collector and 10% of accrued fees to a routing contract on the parent chain. From here, the chain owner has complete control over their earned fees, and the routing contracts can direct AEP fees to a collecting address for the Arbitrum DAO.

ChildToParentRouter

AEP fees from theRewardDistributor must first be sent to Ethereum before they can be deposited to the DAO-controlled address on Arbitrum One. To facilitate this transfer to Ethereum, AEP fees are sent through a series of contracts known asChildToParentRouters.

TheChildToParentRouter is configured to withdraw a single token (immutable and specified at deployment) from the child chain to a specific target address on the parent chain: either anotherChildToParentRouter or theParentToChildRouter on Ethereum.

ParentToChildRouter

All AEP fees from the network of Orbit chains will arrive at theParentToChildRouter on Ethereum. This contract can send ETH and arbitrary ERC-20 tokens to a DAO-controlled address on Arbitrum One.

Deploying your AEP fee router contracts

An Orbit chain is responsible for deploying allChildToParentRouters necessary for their AEP funds to arrive at theParentToChildRouter on Ethereum.

This includes:

- Deploying aChildToParentRouter
- on their Orbit chain configured for their gas token and configured to send funds to either:* TheParentToChildRouter
  - on Ethereum (assuming the network is a Layer-2)
  - AnotherChildToParentRouter
  - configured to the same gas token and configured to send funds to a successive parent chain (this is the case for a Layer-3 network or higher)
- Deploying aRewardDistributor
- contract configured to forward 10% of fees to theChildToParentRouter
- and 90% to the chain owner's preferred reward-receiving address.

In the event that aChildToParentRouter does not connect to theParentToChildRouter on Ethereum, an Orbit chain must deploy successiveChildToParentRouter contracts until a connection toParentToChildRouter is established.

And optionally, an Orbit chain can decide to deduct assertion costs by following the instructions in the[Deducting Assertion Costs](#) section:

Layer-3 chains with custom gas tokens with L2-based token contracts cannot send their custom gas tokens to Ethereum. A future version of the AEP Fee Router may allow Orbit chains with L2-based tokens to distribute fees using the routing system.

In the absence of these, please sendETH through the AEP Fee Router to fulfill your AEP license obligations.

Deployment scripts

The Orbit SDK provides a[configurable script](#) that allows a chain operator to deploy quickly and set up the AEP fee router contracts.

⚠ The standard script deploys and sets up the AEP fee router contracts for L2 chains. For L3 chains (or further layers), a different target address on the parent chain will need to be specified, depending on the native token of the chain. Please contact the Arbitrum Foundation to confirm the target address to withdraw the AEP fees to. The script performs the following operations:

1. Obtain the Rollup and inbox contract of the chain. These are needed to execute the next steps.
2. Obtain the current fee collectors of the chain: Orbit base fee collector, Orbit surplus fee collector, and Parent chain surplus fee collector.
3. Deploy theChildToParentRouter
4. contract, configured to send the amounts received to the appropriate Arbitrum DAO controlled address on the parent chain.
5. Deploy aRewardDistributor
6. contract for each different fee collector account, configured to distribute 90% of the amounts received to the current fee collector and 10% to the ChildToParentRouter contract deployed in the previous step.
7. Set each of the fee collectors to theRewardDistributor
8. contracts

ℹ Note that if the same address collects all three fee types, only oneRewardDistributor contract will be deployed, which will collect all those fees. To configure the script, you need to specify the following[environment variables](#) :

- ROLLUP_ADDRESS
- : address of the Rollup contract
- CHAIN_OWNER_PRIVATE_KEY
- : private key of the account with executor privileges in theUpgradeExecutor
- admin contract for the chain
- ORBIT_CHAIN_ID
- :chainId
- of the Orbit chain
- ORBIT_CHAIN_RPC
- : RPC of the Orbit chain
- PARENT_CHAIN_ID
- :chainId
- of the parent chain, which can neither be an Arbitrum chain nor Ethereum.
- PARENT_CHAIN_TARGET_ADDRESS
- : address on the parent chain where 10% of the revenue will be sent to. You can find the potential target addresses in the[canonical contracts](#)
- section of this document. If the parent chain your chain settles to is not on that list, contact the Arbitrum Foundation to obtain a specific target address for your chain.

Finally, follow these steps to execute the script (from theexamples/setup-aep-fee-router folder):

1. Install dependencies

yarn

install 1. Create .env file and add the env vars

cp .env.example .env 1. Run the script

yarn dev [Edit this page](#) Last updatedonNov 4, 2024 [Previous AEP fee router: overview](#) [Next Calculating AEP license fees](#)