

Enable fast withdrawals on your Orbit chain

Optimistic Rollups must sustain a multi-day challenge period to allow time for fraud proofs. This delays finality for users and dApps, resulting in multi-day withdrawal times and cross-chain communication delays.

Fast withdrawals is a new configuration allowing Orbit chains to achieve fast finality. When an Orbit chain operates on Fast Withdrawals, its transactions will be processed by a committee of validators. Transactions reaching a unanimous vote across the committee will be immediately confirmed.

This will allow:

- Setting up a withdrawal frequency of any time period (up to 15 minutes)
- Users' withdrawals confirmation on the parent chain at frequencies up to ~15 minutes
- Cross-chain dApps to read the finalized state at the same rate as the fast withdrawal frequency

Recommended configuration

While any Orbit chain can adopt Fast Withdrawals, we only recommend that fast withdrawals be adopted by AnyTrust chains with a minimum validator and DAC member requirement. We explain both these recommendations below:

Fast withdrawals for AnyTrust chains

As AnyTrust chains are an optimum (an optimistic rollup using a separate data availability layer), AnyTrust chains are already placing a trust assumption on their Data Availability Committee (DAC) to provide the data needed for fraud proofs and recreating the chain.

The optimal setup for an AnyTrust chain is to have all DAC members also run validators as part of the fast withdrawals committee. This will leverage the existing trust assumption placed on the DAC operators such that enabling fast withdrawals does not add any new trusted parties.

It is possible for an Orbit Rollup to adopt fast withdrawals. However, it would technically no longer be a Rollup as the minimum trust assumption will shift to the trust placed in the Fast Confirmations committee.

Minimum validator and DAC nodes

We recommend that any Fast Withdrawals-enabled chain have at least three DAC members and three validators acting in the fast withdrawals committee. Given that fast withdrawals will enable confirmation of new Rollup state much faster than the usual 7-day challenge period (15 minutes for L2s and 15 seconds for L3s), it becomes even more important to have additional parties involved in validation to further reduce trust assumptions. This requirement can be met with three total operators, who each run a single DAS node and a single validator.

Technical lower bound for fast withdrawals

Once fast withdrawals is enabled, the committee will confirm transactions at the configured frequency. However, a higher network load can cause the fast withdrawals committee to experience slight delays from the configured rate.

- For low-to-medium activity chains (< 1 Mgas/s), 15 seconds is considered to be the sustained lower bound for Fast Withdrawals.
- For chains with higher throughput (>1 Mgas/s), the practical lower bound for fast withdrawals is between 1-2 minutes.

Chain owners and operators should be aware that the fast withdrawals committee may take longer to confirm new assertions under conditions with greater network load. This behavior is to be expected and does not interfere with the security or trust model of the fast withdrawals committee.

Practical lower bounds concerning parent chain finality

While a fast withdrawals-enabled chain can be configured to finality in as little as 15 seconds, there are externalities on the parent chain and from cross-chain messaging layers that must be considered.

For an Ethereum-based Layer-2, we recommend that the fast withdrawal frequency remain above 12.8 minutes, which is the time for Ethereum to achieve finality. For non-Ethereum L1s, we similarly recommend staying above the accepted finality threshold specific to that L1.

For an Arbitrum One-based Layer-3, there are three tiers of finality to consider:

1. Soft finality from the sequencer's confirmation of transaction inclusion (~250ms)
2. Safe finality from batch inclusion after Arbitrum One's assertion is included in an Ethereum block.

3. Hard finality after the Ethereum block containing Arbitrum One's batch is finalized on Ethereum (~15 minutes)

Layer-3 Orbit chains can make their own determination about what level of finality to accept. We consider it safe to rely on soft finality, which would practically enable an Orbit chain to configure fast withdrawals down to 15 seconds.

Adoption instructions (example script)

To enable the fast withdrawals feature, there are three actions to take:

1. Make sure the chain is using nitro-contracts v2.1.0 or above
2. Activate the fast withdrawals feature
3. Upgrade the node software to nitro v3.1.2 or above

Upgrading to nitro-contracts v2.1.0

As mentioned above, the fast withdrawals feature is available for chains that are using nitro-contracts v2.1.0 or above, especially theRollupAdminLogic and theRollupUserLogic contracts. You can check what nitro-contracts version your chain is using by running the[orbit versioner script](#).

If your chain is not running with nitro-contracts v2.1.0 or above, you'll need to perform an upgrade to enable this version. The[orbit versioner script](#) will provide the upgrade paths needed to reach v2.1.0, but basically:

- If the chain is running nitro-contracts v1.1.x, you need to[upgrade first to v1.2.1](#)
- .
- If the chain is running nitro-contracts v1.2.1, you need to[upgrade to v2.1.0](#)
- .

Upgrading to the new nitro-contracts version also requires updating the node software. For v2.1.0, validator nodes and the batch poster node should run[nitro v3.1.2](#) or above.

Suppose you're upgrading your nitro-contracts from v1.2.1 to v2.1.0 and using the standard WASM module root (without customizations). In that case, there are[action contracts available in the supported chains](#). If you're using a customized nitro software, with a different WASM module root, you can still deploy the action contract referencing your modified WASM module root (pre and post upgrade).

Activating fast withdrawals

Once the chain runs nitro-contracts v2.1.0 or above, the new fast withdrawal parameters will be available in theRollupAdminLogic and theRollupUserLogic contracts.

Both the Orbit SDK and the Orbit actions repository provide configurable scripts to activate and configure fast withdrawals on an AnyTrust chain. You can use either of those to activate the feature. Both scripts perform the same actions.

i Even though two scripts are available to activate Fast Withdrawals, you only need to execute one of them. Both scripts perform the same actions. Orbit SDK script

The Orbit SDK provides an[example script](#) to set up a fast withdrawal committee by performing the following operations:

1. Create a new n/n Safe wallet with the specified validators as signers
2. Add the specified validators to the Rollup validators allowlist
3. Set the new Safe wallet as theanytrustFastConfirmer
4. in the Rollup contract
5. Set the newminimumAssertionPeriod
6. if needed
7. Show how to configure the batch poster and validator nodes

To configure the script, you need to specify the following environment variables:

Variable Name Description CHAIN_OWNER_PRIVATE_KEY Private key of the account with executor privileges in the UpgradeExecutor admin contract for the chain. It will be the deployer of the multi-sig Safe wallet. PARENT_CHAIN_ID ChainId of the parent chain. ROLLUP_ADDRESS Address of the Rollup contract. FC_VALIDATORS Array of fast-withdrawal validators. They will be added as signers to the multisig Safe wallet and to the Rollup's validator allowlist. It is recommended that these are DAC members of the AnyTrust chain. MINIMUM_ASSERTION_PERIOD Optional parameter that defaults to 75 blocks (~15 minutes). Minimum number of blocks that have to pass in between assertions (measured in L1 blocks). Finally, follow these steps to execute the script (from theexamples/setup-fast-withdrawal folder):

1. Install dependencies

yarn

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

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

yarn run dev Orbit actions script

The Orbit actions repository also provides an [action script](#) to activate fast withdrawals by performing the following operations:

1. Make sure the "Validate fast confirmation" has not been enabled yet
2. Create a Safe contract for the fast confirmation committee
3. Set the Safe contract as the fast confirmer on the Rollup
4. Set the Safe contract as a validator on the Rollup
5. SetsetMinimumAssertionPeriod
6. to 1 block to allow more frequent assertion

To configure the action script, you need to specify the following environment variables:

Variable Name Description UPGRADE_ACTION_ADDRESS Address of the upgrade action to execute. A standard version is deployed in all [supported chains](#) . If you need to deploy your own action, execute the first step [of this process](#) . PARENT_UPGRADE_EXECUTOR_ADDRESS Private key of the account with executor privileges in the UpgradeExecutor admin contract on the parent chain. It will be the deployer of the multi-sig Safe wallet. PARENT_CHAIN_RPC RPC endpoint of the parent chain. ROLLUP Address of the Rollup contract. FAST_CONFIRM_COMMITTEE Comma-separated list of fast-withdrawal validators. They must be allowlisted validators in the Rollup contract. They will be added as signers to the multisig Safe wallet. It is recommended that these are DAC members of the AnyTrust chain. Finally, follow these steps to execute the script (from thescripts/foundry/fast-confirm folder):

1. Install dependencies

yarn

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

cp .env.example .env 1. Execute the action. The upgrade can be executed usingcast 2. CLI command [\(cast is a part of the Foundry tools](#) 3.), using the owner account (the one with executor rights on parent chainUpgradeExecutor 4.) to send the transaction:

```
( export ( cat .env |
xargs )
&& cast send PARENT_UPGRADE_EXECUTOR_ADDRESS
"execute(address, bytes)"
UPGRADE_ACTION_ADDRESS
( cast calldata "perform(address, address[])" ROLLUP \ [ FAST_CONFIRM_COMMITTEE \ ] ) --rpc-url
PARENT_CHAIN_RPC
--account EXECUTOR )
```

use --account XXX / --private-key XXX / --interactive / --ledger to set the account to send the transaction from

NOTE: If you have a multisig as executor, you can use the following command to create the payload for calling into thePARENT_UPGRADE_EXECUTOR :

```
( export ( cat .env |
xargs )
&& cast calldata "execute(address, bytes)"
UPGRADE_ACTION_ADDRESS
( cast calldata "perform(address, address[])" ROLLUP \ [ FAST_CONFIRM_COMMITTEE \ ] ) )
```

Configure fast withdrawal on nitro v3.1.2 or above

To be able to use Fast Withdrawal on your chain, the batch poster and the validators of the chain need to be running[nitro](#)

[v3.1.2](#) or above.

The following parameters need to be configured in those nodes.

Batch poster

Option Description `--node.batch-poster.max-delay=0h15m0s` Since batches need to be posted so validators can create and confirm assertions, the maximum delay should be set to an amount close to the `minimumAssertionPeriod` defined in the Rollup contract. Modify `0h15m0s` to the configured value. Validators

Option Description `--node.staker.enable-fast-confirmation=true` Enables fast withdrawals in the validator node --
`node.staker.make-assertion-interval=0h15m0s` Since assertions need to be created for them to be confirmed, the minimum interval to create these assertions should be set to an amount close to the `minimumAssertionPeriod` defined in the Rollup contract. Modify `0h15m0s` to the configured value. [Edit this page](#) Last updated on Oct 30, 2024 [Previous Implement Circle bridged USDC Next AEP fee router: overview](#)