Proof of Reserve Feeds

Chainlink Proof of Reserve Feeds provide the status of the reserves for several assets. You can read these feeds the same way that you read other Data Feeds. Specify the Proof of Reserve Feed Address that you want to read instead of specifying a Price Feed address. See the Using Data Feeds page to learn more.

To find a list of available Proof of Reserve Feeds, see the Proof of Reserve Feed Addresses page.

Types of Proof of Reserve Feeds

Reserves are available for both cross-chain assets and offchain assets. This categorization describes the data attestation variations of Proof of Reserve feeds and helps highlight some of the inherent market risks surrounding the data quality of these feeds.

Reserves are available for both offchain assets and cross-chain assets. Token issuers prove the reserves for their assets through several different methods.

Offchain reserves

Offchain reserves are sourced from APIs through an external adapter .

Offchain reserves provide their data using the following methods:

- Third-party: An auditor, accounting firm, or other third party attests to reserves. This is done by combining both fiat and investment assets into a numeric value that is reported against the token.
- Custodian: Reserves data are pulled directly from the bank or custodian. The custodian has direct access to the bank or vault holding the assets. Generally, this works when the underlying asset pulled requires no additional valuation and is simply reported onchain.
- ⚠ Self-attested: Reserve data is read from an API that the token issuer hosts. Self-attested feeds carry additional risk.

Cross-chain reserves

Cross-chain reserves are sourced from the network where the reserves are held. Chainlink node operators can report cross-chain reserves by running an external adapter and querying the source-chain client directly. In some instances, the reserves are composed of a dynamic list of IDs or addresses using a composite adapter.

Cross-chain reserves provide their data using the following methods:

- Wallet address manager: The project uses the IPoRAddressList wallet address manager contract and self-attests to which addresses they own.
- Wallet address: The project attests which addresses they own through a self-hosted API.

Using Proof of Reserve Feeds

Read answers from Proof of Reserve Feeds the same way that you read other Data Feeds. Specify the Proof of Reserve Feed Address that you want to read instead of specifying a Price Feed address. See the Using Data Feeds page to learn more.

Using Solidity, your smart contract should reference AggregatorV3Interface , which defines the external functions implemented by Data Feeds.

```
// SPDX-License-Identifier:
MITpragmasolidity^0.8.7;import{AggregatorV3Interface}from"@chainlink/contracts/src/v0.8/shared/interfaces/AggregatorV3Interface.sol";contractReserveConsumerV3{AggregatorV3Interfaceinternalrese
* Network: Ethereum Mainnet * Aggregator: WBTC PoR * Address: 0xa81FE04086865e63E12dD3776978E49DEEa2ea4e */constructor()
{reserveFeed=AggregatorV3Interface(0xa81FE04086865e63E12dD3776978E49DEEa2ea4e);}/ * Returns the latest price /functiongetLatestReserve()publicviewreturns(int){// prettier-ignore(/uint80
roundID/,intreserve,/uint startedAt/,/uint timeStamp/,/uint80 answeredInRound*/)=reserveFeed.latestRoundData();returnreserve;}}
```

Open in Remix What is Remix? Disclaimer

Proof of Reserve feeds can vary in their configurations. Please be careful with the configuration of the feeds used by your smart contracts. You are solely responsible for reviewing the quality of the data (e.g., a Proof of Reserve feed) that you integrate into your smart contracts and assume full responsibility for any damage, injury, or any other loss caused by your use of the feeds used by your smart contracts.

Learn more about making responsible data quality decisions.