# Oracle Deployment Guide

This section provides a comprehensive guide on deploying an oracle, including the nuances of script parameters and additional setup steps. Follow these steps to ensure accurate and reliable oracle deployment for price data retrieval.

## Step 1: Surveying DEX Liquidity

1. Identify DEXes with Sufficient Liquidity: Begin by surveying the network for Decentralized Exchanges (DEXes) that offer sufficient liquidity. This ensures the oracle can retrieve reliable and accurate price data.

## Step 2: Selection of DEXes

2. Select Supported DEXes: Choose DEXes that are supported by `SpotPriceAggregator` or are forks of supported protocols. Supported DEXes can be found in the `contracts/oracles/` directory of the project.

## Step 3: Network Configuration

3. Configure the Network Settings:
   - Skip this step if your network is supported. This can be checked by observing whether the network is mentioned (registered or not) during a test run, visible in the console output. This verification can be done also by reviewing the `registerAll` method in the `Networks` class. If your network is listed there, it's considered supported, and no further action is required for registration in this step.
   - Update the Hardhat settings file to configure the network.
   - Utilize the `Networks` class from solidity-utils for network registration.
   - Example configuration snippet:

     ```
     ...
     const { Networks } = require('@1inch/solidity-utils/hardhat-setup');
     const net = new Networks(true, 'mainnet', true);
     net.register(your_network_name, networkId, process.env.YOURNETWORK_RPC_URL, process.env.YOURNETWORK_PRIVATE_KEY, etherscan_network_name, process.env.YOURNETWORK_ETHERSCAN_KEY);
     const networks = net.networks;
     ```

```
const etherscan = net.etherscan;
...
```

## Step 4: Environment Variables

4. Set Environment Variables: Define necessary environment variables in the `.env` file located at the project root. Include variables such as `YOURNETWORK_RPC_URL`, `YOURNETWORK_PRIVATE_KEY`, and `YOURNETWORK_ETHERSCAN_KEY` with appropriate values:
   - `YOURNETWORK_RPC_URL`: The RPC URL for accessing your network's node. This URL can support the HTTP header 'auth-key'. To use this header, append the header value to the URL using the `|` symbol. For example: `http://localhost:8545|HeaderValue`. This format allows you to authenticate requests to your node.
   - `YOURNETWORK_PRIVATE_KEY`: Your account's private key, which should be entered without the `0x` prefix. This key is used for deploying contracts and executing transactions on the network.
   - `YOURNETWORK_ETHERSCAN_KEY`: The API key for an Etherscan-like blockchain explorer that supports your network. This key is necessary for verifying and publishing your contract's source code. Ensure you register for an API key with a compatible explorer service for your network.

## Step 5: Deploying Oracles

5. Deploy Oracles:
   - Use the deploy script located at `deploy/commands/simple-deploy.js`. You can find a description of the script and how to use it in the scripts description.
   - Configure the `PARAMS` object for each protocol you wish to deploy an oracle for. The parameters include:
     - contractName: Name of the contract from the `contracts/oracles/` directory.
     - args: Arguments required by the contract (See contract's constructor).
     - deploymentName: A name for your deployment, which will be used to create a file in the `deployments/` directory.
   - Ensure the `skip` flag is set to `false` to proceed with deployment.
   - Example command for deployment: `yarn && yarn deploy <your_network_name>`.

## Step 6: Deploying Wrappers

6. Deploy Wrappers:
   - Follow similar steps as step 5 to deploy necessary wrappers and `MultiWrapper`. You can find different wrappers in the `contracts/wrappers/` directory. After `MultiWrapper` is deployed, it will be possible to edit these lists of wrappers.

## Step 7: Deploying OffchainOracle

7. Deploy OffchainOracle:

- Follow similar steps as step 5 to deploy the `OffchainOracle`. Make sure to include the deployed oracles (from step 5), `MultiWrapper` with wrappers (from step 6) and specifying the tokens you wish to use as connectors for price discovery. After `OffchainOracle` is deployed, it will be possible to edit these lists of oracles and connectors.

Previous
‹ Prices for requested tokens

Next
Introduction ›