

Debugging and Troubleshooting Upkeeps

Given an upkeep ID, this page contains different methods of understanding and fixing issues with Upkeeps.

[Automation debugging script](#)

You can use the [Automation debugging script](#) to debug and diagnose possible issues with registered upkeeps in Automation 2.1 registries. The script can debug custom logic upkeeps, log trigger upkeeps, and upkeeps that use StreamsLookup.

[Underfunded upkeeps](#)

In the Chainlink Automation app, you can see the registration details of the upkeep, alongside the balance required and performUpkeep history. If the upkeep is underfunded, you will see a warning on top of the page. Underfunded upkeeps will not be performed.

[Insufficient perform gas](#)

If your performGasLimit is too low, the Automation Network will not execute your upkeep as it will fail in simulation. Consider increasing your perform upkeep gas limit in the UI. See [supported Blockchain Networks](#) for limits.

[Insufficient check gas](#)

If the amount of computation in your checkUpkeep exceeds our checkGasLimit, your upkeep will not be performed. You will have to reduce the amount of compute in checkUpkeep to bring the gas below the applicable limits. See [supported Blockchain Networks](#) for limits.

[Paused upkeeps](#)

If your upkeep is paused, your upkeep will not be performed. Please unpause it in the [Chainlink Automation app](#).

[StreamsLookup](#)

[Upkeep has not been allowlisted](#)

Once you registered your upkeep, you need to [ask the Data Streams team to allowlist your upkeepID](#) and specify the feeds you will need to access. If your upkeepID has not been added to the allow list it will not perform an upkeep.

[Requesting multiple feeds where one is not valid](#)

It is possible to request multiple feeds by specifying the feeds in a string array. However, if one of the reports is invalid or not available then Automation will not return any values to your checkCallback function. This is to ensure correct execution of your contract.

[Etherscan](#)

You can view the registry or user's upkeep address or [Etherscan](#) to view transaction history. There are three types of information you can find on Etherscan:

- All transactions on registry: This shows all the perform upkeeps on the registry. You can view a specific performUpkeep transaction to debug more.
- Specific performUpkeep transaction: By diving deep in the logs, you can check the upkeep ID within the UpkeepPerformed log.
- Target's internal transactions: For a particular target contract, you can view its internal transactions which contains performUpkeep transactions for that contract by using the registry address as the filter from address. Note: internal transactions are only visible on Etherscan.

[Tenderly](#)

You can use [Tenderly](#) to simulate checkUpkeep and/or performUpkeep on different blocks. Before following the steps below, make sure you have a Tenderly account.

1. Enter the address of your selected registry. You can find this on the [Supported Networks](#) page.
2. Select your network.

3. Click `Fetch ABI` to automatically fetch the registry ABI.
4. Select the `checkUpkeep` function or `performUpkeep` function.
5. Enter the ID of your Upkeep. You can find this in the [Chainlink Automation app](#).
6. You can either enter a block number to simulate a past action or use a pending block number to view the current state of an action and view the end result of an of an action.
7. Once the simulation is complete, you will see the result. This will be either a `success` or an `error`. To understand errors, view information under the `Debug` tab. Note: if the `performUpkeep` is failing while the check is succeeding, Chainlink Automation will not broadcast transactions.