

## Etherscan Guide:

### Browse:

<https://etherscan.io/>

### Looking at transactions-

Select a contract address....and notice the list of transactions with a from and to address. These are the wallet addresses from and to parties

### Looking at accounts:

Go to:

**<https://etherscan.io/accounts>**

Click on any of the top account addresses....they will be listed by amount of ether

Notice the columns-

Balance

% ownership

# transactions

Select and view transactions

### Verify and Publish:

<https://medium.com/coinmonks/how-to-verify-and-publish-on-etherscan-52cf25312945>

Use flattener below and paste into the verify box

## The truffle flatterer:

Install globally:

## Installation

```
npm install truffle-flattener -g
```

## Usage

Just install it with npm in your truffle project and run truffle-flattener <solidity-files>.

This tool helps you to verify contracts developed with Truffle on [Etherscan](#), or debugging them on [Remix](#), by merging your files and their dependencies in the right order.

```
npm install --save-dev truffle-flattener
```

Next, we'll use the [truffle-flattener](#) tool to build a single file of text.

## How to use locally:

```
truffle-flattener ./contracts/GreatestShow.sol > ./GreatestShowFlattened.sol
```

As above...go to directory where the contract lives...

Run:

```
truffle-flattener contractName > outputFileName
```

You can also Flatten then deploy on Remix

## **Verify an ICO:**

**<https://blocksplain.com/2018/02/20/etherscan/>**

Also note the Contract Address. An above-board ICO will be performed on a public blockchain (in this case, Ethereum) and the contract address should be published. That allows external people – and the media – to check and verify the details of the ICO. Again, some ICOs either don't do their token event on a public blockchain, or they do not tell us their contract address. Either case makes it very difficult to verify the amount they claimed to raise.

## **Token Transfers:**

Now let's go to "Token Transfers" and click the "Last" button (on the right) to see the first lot of transactions, which were the opening buys of AGI in the token sale. If you scan through the last five or so pages, you'll see the number 36,750 recur. Since there's nothing over that number, that must've been 5 ETH worth of AGI. So the price of ETH at that time, at least for the purposes of purchasing AGI, was US\$735.

If you keep scanning through the pages, you'll see that the transactions seem to match up to what the company promised to sell.

Although checking Etherscan isn't a guarantee that an ICO made what they claimed, because other numbers can always be fudged, it's a way to check the transaction flow. If that looks good and other aspects of the ICO pass muster – see [Tips for assessing an ICO](#) – then Etherscan helps to verify the company.

In this case, I'm pretty confident that SingularityNET sold what they claimed and in the manner they outlined (i.e. maximum of 5 ETH per user).

The ICOs to be wary of are the ones where you **cannot** find solid data about what they supposedly raised. It's all about transparency and if an ICO doesn't disclose their transaction data, then something fishy is probably going on.