

# Arweave Upload Guides

In order to facilitate a permissionless proposal process before on-chain governance is reimplemented in Beanstalk, proposers must upload verified message signatures to Arweave. Anyone can use these to verify that the Proposer Wallet has at least 0.1% of total Stalk at the beginning of the Voting Period (and at least 0.1% of total Voting Stalk at the end of the Voting Period).

- [BIP Proposer Wallet Format](#)
- [BOP Proposer Wallet Format](#)

BCM Signers must upload verified message signatures to Arweave confirming that they've verified a given multisig transaction according to the processes approved by the DAO. This limits blind signing and encourages each Signer to independently verify that a proposed transaction is accurately represented. Anyone can verify that a Signer verified a proposed transaction during the Voting Period.

- [BIP Verification Approval Format](#)
- [EBIP Verification Approval Format](#)
- [BIR Verification Approval Format](#)
- [Verification Rejection Format](#)

Check out the guides on setting up an Arweave wallet, creating verified message signatures and uploading them to Arweave:

- [Creating an Arweave Wallet](#)
- [Creating an Etherscan Verified Message](#)
- [Uploading a File to Arweave](#)

## BIP Proposer Wallet Format

The proposer of a BIP must include an Arweave upload of a JSON file in the **Proposer** section of the written proposal that confirms the Proposer Wallet of the BIP.

The JSON file should use the following format:

```

{
  "address": "[address]",
  "signed_message": "Confirming that this address is the Proposer Wallet
for BIP-[x] with commit hash [hash].",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
  // An array where each element has transaction data to execute
  "transaction_data": [
    {
      // The order for the BCM to execute the function calls
      "call_order": 0,

      // The owner function to be called by the BCM,
      // followed by each input parameter
      "function": "[function name]",
      "[input parameter 1]": "input parameter 1 value",
      ...
      "[input parameter n]": "input parameter n value"
    },
    ...
    {
      "call_order": n,
      "function": "[function name]",
      "[input parameter 1]": "input parameter 1 value",
      ...
      "[input parameter n]": "input parameter n value"
    },
  ],
}

```

Example:

- Arweave: [https://arweave.net/jCd\\_pfSUzjz6ZOJfhV0xp9bnDlogWswvZvL6SK4QB2A](https://arweave.net/jCd_pfSUzjz6ZOJfhV0xp9bnDlogWswvZvL6SK4QB2A)
- Etherscan: <https://etherscan.io/verifySig/12872>

## BOP Proposer Wallet Format

The proposer of a BOP must include an Arweave upload of a JSON file in the **Proposer** section of the written proposal that confirms the Proposer Wallet of the BOP.

The JSON file should use the following format:

```
{
  "address": "[address]",
  "signed_message": "Confirming that this address is the Proposer Wallet for
BOP-[x].",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
}
```

Example:

- Arweave: <https://arweave.net/MPLhmrlfR3YIHacpNKkyNyHe8SSYdZjbMabeO47h-mc>
- Etherscan: <https://etherscan.io/verifySig/12871>

## BIP Verification Approval Format

Every BCM Signer is expected to know how to verify `di amondCut` data and confirm they have verified transactions by uploading signed messages to Arweave.

The JSON file should use the following format:

```
{
  "address": "[address]",
  "signed_message": "Confirming that I have reviewed BIP-[X]
with BCM transaction nonce [Y], Safe transaction hash [tx hash],
commit hash [hash], GitHub PR [URL] and Snapshot proposal [URL].",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
}
```

Example:

- Arweave: <https://arweave.net/GatLxpVuazQoczf2rRaciWepZBzzZfme5yKBmHFbsko>
- Etherscan: <https://etherscan.io/verifySig/12876>

## EBIP Verification Approval Format

Every BCM Signer is expected to know how to verify `di amondCut` data and confirm they have verified transactions by uploading signed messages to Arweave. *A verified message signature is not required in cases where functions are only being removed from Beanstalk.*

The JSON file should use the following format:

```
{
  "address": "[address]",
  "signed_message": "Confirming that I have reviewed EBIP-[X]
with BCM transaction nonce [Y], Safe transaction hash [tx hash],
commit hash [hash], GitHub PR [URL] and Arweave proposal [URL].",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
}
```

Example:

- Arweave: [https://arweave.net/MHImrY\\_D5GsxfN0\\_kPAhJCGmjPvkCyQGBAYI8Cf32KY](https://arweave.net/MHImrY_D5GsxfN0_kPAhJCGmjPvkCyQGBAYI8Cf32KY)
- Etherscan: <https://etherscan.io/verifySig/12875>

## BIR Verification Approval Format

Every BCM Signer is expected to know how to verify `di amondCut` data and confirm they have verified transactions by uploading signed messages to Arweave.

The JSON file should use the following format:

```
{
  "address": "[address]",
  "signed_message": "Confirming that I have reviewed BIR-[X]
with BCM transaction nonce [Y], Safe transaction hash [tx hash]
and Snapshot proposal [URL].",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
}
```

Example:

- Arweave: <https://arweave.net/B5IGjHpe0VI44QT0acaHVorO4Tjv7-UUMjiJCMMAHUE>
- Etherscan: <https://etherscan.io/verifySig/12874>

## Verification Rejection Format

The BCM will not submit a transaction that was misrepresented in the Snapshot proposal.

In the case that any Signer during the verification process determines that a Snapshot proposal does not accurately represent the transaction, that Signer will sign a message indicating as such with context on the issue.

The JSON file should use the following format:

```
{
  "address": "[address]",
  "signed_message": "Confirming that I have reviewed BCM transaction
nonce [x] and determined that the transaction has the following issue: [issue]",
  "signature_hash": "[signature hash]",
  // Optional if Etherscan is down
  "etherscan_link": "[verified Etherscan signature link]"
}
```

Example:

- Arweave: [https://arweave.net/cURpVo2mnwxijm7pn19TbxRZsOsIPoBK\\_ASWfM6OcyE](https://arweave.net/cURpVo2mnwxijm7pn19TbxRZsOsIPoBK_ASWfM6OcyE)
- Etherscan: <https://etherscan.io/verifySig/12873>

## Guides

### Creating an Arweave Wallet

- [Official Docs: Generating an Arweave Wallet](#)

### Creating an Etherscan Verified Message

1. Go to <https://etherscan.io/verifiedSignatures>
2. Select "Sign Message" and select "OK" on the alert.
3. Connect your wallet.
4. Insert the message to sign in the "Message to Sign" field and select "Sign Message".
5. Confirm the transaction in your wallet and your hardware wallet, if applicable.
6. Select "Publish".
7. You'll be taken to the URL for your verified signature.

### Uploading a File to Arweave

This is a guide on uploading a file to Arweave from the command line. There are web-based alternatives like [Ardrive](#).

1. Open your terminal.
2. Install the latest version of Arkb: `npm install -g arkb`
3. Save your keyfile: `arkb wallet-save path/to/key.json`
4. Deploy a file to Arweave: `arkb deploy file.json`

Only step 4 is necessary for future uploads.