Inheritor Emergency Management Tool

User Manual

Table of Contents

Introduction

The Inheritor Emergency Management Tool is designed as a failsafe mechanism that provides testators direct control over their digital inheritance smart contracts. This tool should only be used in emergency situations when normal access methods are unavailable, or when you need to perform critical operations directly.

With this tool, you can:

- View your Digital Will information (all inheritances and check-in status)
- Perform a check-in to reset inheritance timers
- Revoke inheritances to permanently cancel them
- Remove verifiers from your Digital Will
- Return unused ETH to your gas wallet

This tool provides a command-line interface for emergency management of your digital inheritances without requiring the full mobile application.

Before You Begin

Before using this tool, you'll need:

1. **Recovery Mnemonic**: Your 12 or 24-word recovery phrase for the testator account
2. **Gas Wallet Private Key**: A private key for a wallet containing ETH to pay for transaction fees
3. **Network Information**: Knowledge of which network (Ethereum or Arbitrum) your Digital Will is on
4. **Internet Connection**: Access to the Ethereum or Arbitrum networks

Installation

Installing Node.js (Required)

This tool requires Node.js, which is NOT included by default in macOS, Windows, or Linux systems. You'll need to install it first:

macOS:

1. Option 1: Download the installer from Node.js website
2. Option 2: If you have Homebrew, run: `brew install node`

Windows:

1. Download and run the installer from Node.js website

Linux:

1. Ubuntu/Debian: `sudo apt update && sudo apt install nodejs npm`
2. Fedora: `sudo dnf install nodejs`
3. Arch: `sudo pacman -S nodejs npm`

Setting Up the Tool

Once Node.js is installed:

1. Save the script as `Testator.js`
2. Open a terminal and navigate to the directory containing the script
3. Install required dependencies:

```
npm install ethers bip39 @ethersproject/hdnode
```

4. Make the script executable (macOS/Linux only):

```
chmod +x Testator.js
```

5. To verify Node.js is installed correctly, run:

```
node --version
```

This should display the Node.js version (should be 16.0.0 or higher)

Running the Tool

To start the tool, run:

```
node Testator.js
```

Initial Setup Process

1. **Enter Testator Recovery Phrase**:
   - Type your complete mnemonic with all words separated by spaces
   - Example: `word1 word2 word3 ... word12`
   - Press Enter after entering all words
   - The script will display the derived address
2. **Enter Gas Wallet Private Key**:
   - Type or paste the private key of your gas wallet (with 0x prefix)
   - Example: `0x123abc...`
   - This wallet must contain ETH for transaction fees
   - The script will display the wallet address
3. **Select Network**:
   - Type either `ethereum` or `arbitrum` (case insensitive)
   - Press Enter to confirm
4. **RPC Configuration**:
   - Choose option `1` if you have your own RPC URL (Infura, Alchemy, etc.)
   - Choose option `2` to use public endpoints
   - If using option `1`, enter your complete RPC URL when prompted
5. **Main Menu**:
   The tool will connect to the network and display the main menu:

```
=== Main Menu ===
1. View Digital Will
2. Revoke all inheritances
3. Check-in
4. Remove verifier
5. Refund remaining ETH to gas wallet
6. Exit
```

User Interface Tips

- **Menu Selection**: Enter only the number (1-6) of your chosen option
- **Yes/No Questions**: Always type the full word `yes` or `no` when prompted
- **Waiting for Transactions**: When sending transactions, be patient while waiting for confirmations
- **Returning to Menu**: After completing an action, press Enter to return to the main menu
- **Viewing Long Output**: Scroll up in your terminal to review detailed information if needed

Understanding Each Function

1. View Digital Will

This function fetches and displays all your inheritance information and check-in status.

- **Process**:
  - Connects to the blockchain
  - Retrieves all inheritance records associated with your address
  - Shows check-in information (last check-in, interval, next due date)
  - Lists all active inheritances with details
- **Display Information**:
  - Check-in details (timing and deadlines)
  - Inheritance IDs
  - Current state (color-coded for easy recognition)
  - Beneficiary addresses
  - Grace periods
  - Scheduled transfers (if any)
- **When to Use**:
  - To verify the current status of all your inheritances
  - To check when your next check-in is due
  - To identify which inheritances you may want to revoke

2. Revoke All Inheritances

This function permanently cancels active inheritances.

- **Process**:
  - Confirms your intention (requires typing "yes")
  - Retrieves all your inheritances
  - Identifies which can be revoked (Designated state only)
  - Funds your testator wallet from your gas wallet if needed
  - Sends revocation transactions for each eligible inheritance
- **Important Notes**:
  - Only inheritances in the "Designated" state can be revoked
  - The tool will automatically skip inheritances in other states
  - Multiple transactions will be sent (one per inheritance)
  - Requires sufficient ETH in your gas wallet
  - This action is permanent and cannot be undone
- **When to Use**:
  - In an emergency situation where you need to cancel pending inheritances
  - If you want to recreate inheritances with different parameters
  - If circumstances have changed and you no longer wish assets to be transferred

3. Check-in

This function resets the timer on all your inheritances.

- **Process**:
    - Confirms your intention (requires typing "yes")
    - Creates a transaction from your testator wallet
    - Transfers ETH from your gas wallet if needed
    - Sends the check-in transaction
    - Waits for confirmation
- **Important Notes**:
    - This resets the timer on all your inheritances
    - Prevents inheritances from becoming claimable
    - Requires sufficient ETH in your gas wallet
    - Must be performed by the testator wallet (handled automatically)
- **When to Use**:
    - To prevent inheritances from becoming claimable
    - When you cannot access the normal app for check-in
    - As a failsafe measure during emergencies
    - To maintain control over your digital assets

4. Remove Verifier

This function removes the verifier requirement from your Digital Will.

- **Process**:
    - Confirms your intention (requires typing "yes")
    - Checks your current verifier status
    - Creates a transaction from your testator wallet
    - Transfers ETH from your gas wallet if needed
    - Sends the transaction to remove the verifier
    - Waits for confirmation
- **Important Notes**:
    - This removes all verification requirements
    - After removal, inheritances will become claimable based solely on check-in timers
    - Requires sufficient ETH in your gas wallet
    - Must be performed by the testator wallet (handled automatically)
- **When to Use**:
    - If your verifier is no longer available or reliable
    - When you want to simplify the inheritance claiming process
    - To remove additional safeguards (use with caution)

- When verification is causing problems with legitimate inheritance transfers

5. Refund Remaining ETH

This function returns unused ETH from the testator wallet to your gas wallet.

- **Process**:
  - Checks the testator wallet's ETH balance
  - Confirms your intention (requires typing "yes")
  - Calculates maximum amount that can be safely refunded
  - Sends transaction to return funds to gas wallet
- **Important Notes**:
  - Small amounts (< 0.001 ETH) cannot be refunded reliably
  - Some ETH is kept to cover the transaction fee
  - Will display both wallets' balances before and after
- **When to Use**:
  - After completing other operations to recover unused ETH
  - When you want to consolidate funds back to your main wallet
  - To avoid leaving small amounts of ETH in the testator wallet

6. Exit

Safely exits the application.

Technical Background

Understanding Key Terms

Recovery Phrase (Mnemonic)

- A series of 12-24 words that generates your private key
- Must be entered with spaces between words
- Extremely sensitive information - never share with anyone
- In the Inheritor system, this recovers your testator EOA address

EOA (Externally Owned Account)

- A standard Ethereum address controlled by a private key
- Derived from your recovery phrase
- Used to identify your Digital Will and inheritances
- The script derives this from your mnemonic automatically

Private Key

- The cryptographic key that controls an EOA

- Used to sign transactions
- For this tool, you need a separate wallet's private key for gas
- Never share private keys with anyone

RPC Provider

- Remote Procedure Call - how the script talks to the blockchain
- Options include Infura, Alchemy, or public endpoints
- Allows reading blockchain data and sending transactions
- Connection may fail with some public endpoints due to rate limiting

Inheritance States

The tool displays inheritance states with color coding for clarity:

- **Designated** (Green): The default state; inheritance is set up but not yet claimable
- **Claimable** (Yellow/Orange): The inheritance can now be claimed by the beneficiary
- **Claimed** (Blue): The inheritance has already been claimed
- **Revoked** (Red): The testator has cancelled this inheritance
- **Purged** (Gray): The inheritance has been removed from the system

Why Funds Transfer is Needed

The Inheritor contract requires transactions to be sent from your testator address, not just any wallet. This creates a unique challenge:

1. Your testator address (derived from your mnemonic) needs to sign transactions
2. This address likely has no ETH to pay for gas fees
3. Your gas wallet has ETH but can't directly perform testator actions

Solution: The tool transfers just enough ETH from your gas wallet to your testator wallet to pay for transactions. This approach:

- Maintains proper contract authorization
- Minimizes the amount of ETH transferred
- Provides an option to return unused ETH afterward

Troubleshooting

Common Errors and Solutions

"Invalid mnemonic phrase"

- **Cause**: Recovery phrase words are incorrect or misspelled
- **Solution**: Double-check each word, ensure correct spacing, and try again

"Invalid private key for gas wallet"

- **Cause**: The private key format is incorrect

- **Solution**: Ensure the key includes the "0x" prefix and contains 64 hexadecimal characters after the prefix

"Connection failed"

- **Cause**: RPC provider is unavailable or rate-limited
- **Solution**: Try option 2 to use public endpoints, or use your own Infura/Alchemy key

"Error getting contract address from proxy"

- **Cause**: Cannot access the proxy contract
- **Solution**: Enter the contract address manually when prompted

"No contract found at address"

- **Cause**: The contract address is incorrect or the contract is not deployed on the selected network
- **Solution**: Verify you're connected to the correct network

"Gas wallet has insufficient funds"

- **Cause**: Not enough ETH in gas wallet to fund the operation
- **Solution**: Add more ETH to your gas wallet address

"Testator wallet still has insufficient funds after transfer"

- **Cause**: The ETH transfer succeeded but price fluctuations made it insufficient
- **Solution**: Try again with a larger transfer amount

"Failed to transfer funds"

- **Cause**: Problem with the ETH transfer transaction
- **Solution**: Check network conditions and try again

"Not authorized: Testator required"

- **Cause**: Transaction not sent from the correct testator address
- **Solution**: This should not occur as the tool handles this automatically

When to Seek Help

If you encounter persistent errors not covered above, please:

1. Take note of the exact error message
2. Do not share your mnemonic or private keys with anyone
3. Contact official support channels

Security Considerations

- **Use on a Secure Device**: Run the tool on a private, secure computer

- **Network Security**: Prefer a trusted network connection
- **Protect Your Mnemonic**: Never share your recovery phrase
- **Private Key Safety**: Your gas wallet private key is sensitive information
- **Temporary Use**: This is an emergency tool, not for regular use
- **Verify Transactions**: Check transaction details before confirming

Related Tools

The Inheritor Emergency Tools suite includes three complementary command-line applications:

1. **Testator Emergency Management Tool** (this tool): For testators to manage their Digital Will
2. **Beneficiary Check Tool**: For beneficiaries to monitor and check claimability of inheritances
3. **Beneficiary Claim Tool**: For beneficiaries to claim and decrypt inherited digital assets

Each tool serves a specific purpose in the inheritance lifecycle. If you are both a testator and a beneficiary, you may need to use different tools depending on your current role.

Disclaimer: This tool is provided for emergency use only. While efforts have been made to ensure its security and accuracy, use it at your own risk. Always verify the effects of any blockchain transactions, as they cannot be reversed once confirmed.