Albacea Script Guide

Installing Bitcoin Core

The Albacea script works on top of the Bitcoin Daemon and bitcoin-cli commands, so in order to use it, we will need to install it correctly on our machine.

Firstly, open a terminal using Ctrl + Alt + T on your keyboard. And type the following command:

```
$ cd /tmp
```

This command will change us to the tmp directory whose files get deleted with every reboot.
Now, we will install the last version of Bitcoin core using wget:

```
$ wget https://bitcoin.org/bin/bitcoin-core-0.21.1/bitcoin-0.21.1-x86_64-linux-gnu.tar.gz
```

This command specifically downloads the version 0.21.1 of Bitcoin, so make sure you update it if the latest version is different.

Now we will unpack the file we just downloaded with:

```
$ tar -xvf <nameOfTheFile>
```

Example:

```
$ tar -xvf bitcoin-0.21.1-x86_64-linux-gnu.tar.gz
```

This command has created a new directory called bitcoin-0.21.1 with all the necessary files.

Now we will install this on the /usr/local/bin path using the following command:

```
$ sudo install -m 0755 -o root -g root -t /usr/local/bin bitcoin-0.21.1/bin/*
```

Finally, to check we did everything correctly, run:

```
$ bitcoind --version
```

The output should be something like:

```
> Bitcoin Core version v0.21.1
```

Installing QR code dependencies

In order to make full use of the script and be able to qr encode the redemption script and public address (and decode them as well), we will need to install the necessary tools

To do it, is as easy as:

```
$ sudo apt install qrencode
```

And:

```
$ sudo apt-get install zbar-tools
```

Let’s try and test it. We will create a qr code that says “Hello”

```
$ qrencode -o test.png “Hello”
```

Finally, to check if we can decode it properly:

```
$ zbarimg test.png
```

The output should be the text we input before.

Creating a multisig wallet

Before running the script we will first need to create a wallet using Bitcoin-Cli. For that:

```
$ bitcoin-cli createwallet <nameOfYourWallet>
```

Example: $ bitcoin-cli createwallet myWallet

Now to make sure everything runs smoothly we will unload and load the wallet

```
$ bitcoin-cli unloadwallet <nameOfYourWallet>
$ bitcoin-cli loadwallet <nameOfYourWallet>
```

What will need to do first is download the Albacea repository from github.
On the terminal, go to the directory you wish (Desktop will do just fine) and download it with this command:

```
$ git clone https://github.com/miguelmartinn9/Albacea
```

This will create a new directory called Albacea with all the files inside it.
Let’s change to this directory then:

```
$ cd Albacea
```

Now to create a new multisig wallet we need to follow the next steps:
(We will be creating a 2of3 wallet, adjust your commands accordingly if you want a different set of keys)

```
$ ./albaceaScript.py entropy --num-keys 3
```

The output will be 3 different sets of computer entropy which you need to save somewhere (just copy it to a text editor)

Now, let’s generate our private keys, public address and redemption script:

```
$ ./albaceaScript.py create-deposit-data -m 2 -n 3
```

We will be asked to roll a dice 62 times and input the results.

After inputting the dice rolls correctly, we will be asked to input some 40 char computer entropy which we have already done (remember the strings we copied to the text editor). You will need to paste one of the strings.
Example: 86c5 c2eb e6a3 a558 690a fa07 a8c0 4d36 fb03 2fc3

You will have to do this as many times as total keys there will be on your wallet.

After doing so, the output will be something like:

Private keys:
Key #1: <private key1>
Key #2: <private key2>
Key #3: <private key3>

Cold storage address:
37kAv8sz7AvGP74oBxYU7Ca3pJXdzFUGkz

Redemption script:
<redemption script>

You must keep all this data in a safe place. The private keys and redemption script can not be lost, or all your funds will be lost. It is recommended not to store it on a digital place (avoid computer files, mobile phones and pictures of it)

Withdrawing from your wallet

To start this process we will run this command:

```
$  ./albaceaScript.py create-withdrawal-data
```

And we will be prompted a series of things to introduce:

Firstly we will be asked for our address, which is the one that was given to us when creating the multi signature wallet.

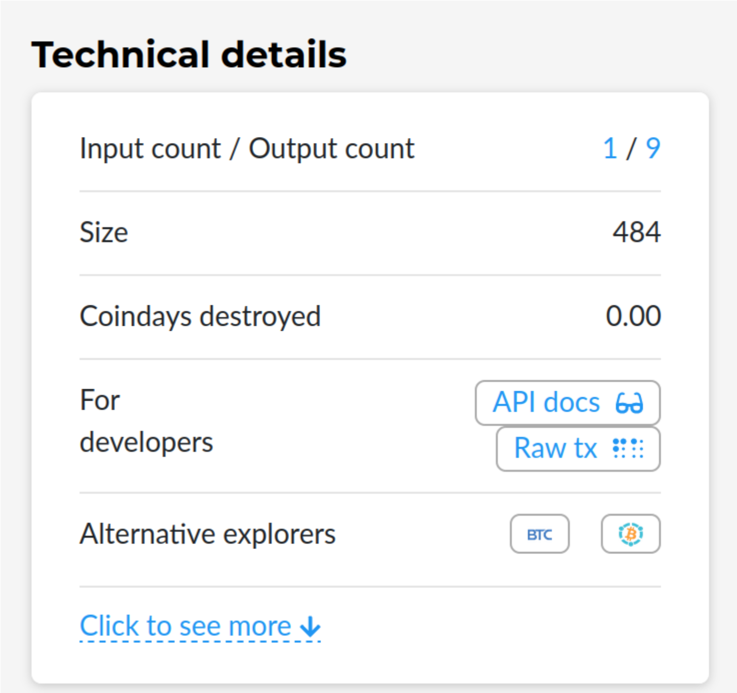
The second thing we will have to input is the Redemption script for the address we just introduced. As you may have guessed, this script is also the one we got when creating the wallet.

Next, the destination address, this is pretty self explanatory.

Now is asking us for the number of Unspent Transaction Outputs (UTXO) that we will be using to make our withdrawal.

An address can contain several coins (UTXO) and we must choose which ones we want to spend.

After selecting the number of UTXO's we will be asked to input the raw transaction data, which we can find in https://blockchair.com/bitcoin/transaction/<TheTransactionIDOfYourUTXO>



Technical details

Input count / Output count	1 / 9
Size	484
Coindays destroyed	0.00
For developers	
Alternative explorers	

Here you will select "Raw tx" and copy the chunk of hexadecimal data and paste it on the terminal.

Now we will be asked by the number of private keys we will be signing the transaction with. This depends on the type of wallet that we created previously. The one we used as an example was 2of3 so we will be using a minimum of 2 private keys.

Now we input the private that we want to sign with.

After this, we will be asked which fee rate we want to select. Depending on our time preference we will want a higher or lower rate, so to be sure just look the current fees on https://mempool.space/

Now we will input the number of Bitcoin we want to withdraw (we can leave it blank, pressing enter, and it will send everything on that UTXO).

Finally, we are given the raw signed transaction ready for broadcasting.

To broadcast it, we have several options:

https://coinb.in/#broadcast
https://bscscan.com/pushTx
https://www.letmegooglethat.com/?q=push+raw+signed+transaction

Check balance and maintenance

To check your wallet's balance you only need to have your public address.
To see your transactions and balance just choose a block explorer of your preference (or even with your own node) and you will be able to check it.

To maintain your wallet, make sure every 6 months or so (time spam really depends on you) to check that:

1. No private keys are lost
2. Your balance is as expected
3. You can withdraw your money (you can withdraw a little to test)

If private keys are lost, you must create a new multi signature wallet and send all your funds there.