

# Abelian CLI Wallet Manual

---

This document describes how to run an Abelian CLI wallet.

Please refer to *Abelian Software Basics* to understand the software architecture first.

The official Abelian download page is located at <https://abelian.info/downloads>. To interact with other miners, developers and users, visit the official Discord server at <https://discord.gg/cPUhCmsw>.

All the operations are carried out via the Command Line Interface (CLI). For Linux or macOS, open Terminal; and for Windows, open PowerShell or any of your favorite shell application.

**Note:** For upgrading from an old version to the latest one, please jump directly to section, **9. Upgrade to the Latest Version**.

## 1. Installation

---

Download two compressed files, something like **abec-linux-amd64-v0.11.9.tar.gz** and **abewallet-linux-amd64-v0.11.9.tar.gz** for Linux running on an x86 architecture (e.g. Intel chips).

Unzip them and put them under the same folder **~/abel/**:

1. `~/abel/abec-linux-amd64-v0.11.9/`
2. `~/abel/abewallet-linux-amd64-v0.11.9/`

There are three executable files: **abec** and **abectl** are in `~/abel/abec-linux-amd64-v0.11.9/` and **abewallet** in `~/abel/abewallet-linux-amd64-v0.11.9/`.

Go to `~/abel/abec-linux-amd64-v0.11.9/`, and run

1. *# macOS and Linux*
2. `./start_abec.sh`
3. *# Windows*
4. `abec`

Then press control+C to stop it. By doing this, we create a **configuration folder** for **abec** located at:

- Windows: `C:\Users\[username]\AppData\Local\Abec`
- macOS: `/Users/[username]/Library/Application Support/Abec`
- Linux: `/home/[username]/.abec`

Next, go to `~/abel/abewallet-linux-amd64-v0.11.9/`, and run

1. *# macOS and Linux*
2. `./start_abewallet.sh --create`
3. *# Windows*

```
4. abewallet --create
```

Then press control+C to stop it. By doing this, we create a **configuration folder** for **abewallet** located at:

- Windows: C:\Users\username\AppData\Local\Abewallet
- macOS: /Users/username/Library/Application Support/Abewallet
- Linux: /home/username/.abewallet

**Note:** On macOS (M1/M2 chip / arm64), if it says `'xxx' is damaged and can't be opened`, run the command

```
xattr -d com.apple.quarantine path/to/xxx .
```

If it says `'xxx' cannot be opened because the developer cannot be verified`, go to `System Preferences -> Security & Privacy -> General` and click `allow anyway`.

## 2. Create a wallet

To create a wallet, go to `~/abel/abewallet-macos-amd64-v0.11.9/` and run

1. *# macOS and Linux*
2. `./start_abewallet.sh --create`
3. *# Windows*
4. `abewallet --create`

Here is an example.

1. `./start_abewallet.sh --create`
2. **Enter** the **private** passphrase **for** your **new** wallet:
3. **Confirm** passphrase:
4. **Enter** the **public** passphrase **for** your **new** wallet:
5. **Confirm** passphrase:
6. NOTE: **Use** the `--walletpass` option to configure your **public** passphrase.
7. **Do** you have an existing wallet seed you want to **use**? (n/**no**/y/yes) [**no**):
8. **Your** wallet generation seed **is**:
9. `afe0f025646cde3eee099db9215f8cdb212ed0e06730fd0087e9d1ff5565fd53`
10. the crypto version **is** 0
11. **Your** wallet mnemonic list **is**:
12. `quiz,always,announce,silver,social,buyer,return,crisp,rice,april,tobacco,rent,base,half,light,toward,wonder,aerobic,whip,physical,vocal,real,vocal,leg`
13. IMPORTANT: **Keep** the version **and** seed **in** a safe place **as** you
14. will NOT be able to restore your wallet without it.
15. **Please** keep **in** mind that anyone who has access
16. to the seed can also restore your wallet thereby
17. giving them access to all your funds, so it **is**
18. imperative that you keep it **in** a secure location.

19. **Once** you have stored the seed **in** a safe **and** secure location, enter "OK" to **continue**: OK
20. **Creating** the wallet...
21. **Please** remember the initial address:
22. 0000000005a38589d86427698e3ec735335b368899ed6e0239c4285bbc4e370f4ea4e6d2ac1f1555b53f8df7c30e13d4bccc3b6d56763ec279620d9f131fb68089cb8ef18885950f84e56bf78d1780a5cac57d0888dabd669f86f85e7055afabae6a332fa000b3c6ee6a09751ce41ad7de4e...

The **public passphrase** will be used when running abewallet, and the **private passphrase** will be used for unlocking the wallet (so to transfer ABEL tokens). The **initial address** will be used as your mining address or payment address later.

**Important:** Keep the **mnemonic list** in a safe place. The mnemonic list can be used for recovering the wallet, install another mining machine while using the same wallet, or upgrade the mining software.

## 3. Configuration - abec

---

Go to the configuration folder of **abec** and make changes as below in **abec.conf**.

You can let other Abelian nodes on the mainnet know your presence through broadcasting your IP address (e.g. 1.2.3.4). This works only if you have a public IP or have some IP forwarding configured at your router:

```
1. externalip=1.2.3.4
```

Allow **abelminer** to connect to **abec** for GPU mining:

```
1. enablegetwork=1
```

If you have multiple GPU mining machines connecting remotely to this single **abec**, make sure that port 8668 is opened:

```
1. rpclistengetwork=:8668
```

Set the initial address of your wallet to miningaddr:

```
1. miningaddr = [your initial address]
```

## 4. Configuration - abewallet

---

Copy the RPC values, namely **rpcuser** and **rpcpass**, from **abec.conf**:

```
1. rpcuser = [rpcuser]
```

```
2. rpcpass = [rpcpass]
```

Paste them respectively to **abewallet.conf** in the configuration folder of **abewallet**.

```
1. abecrpcuser= [rpcuser in abec.conf]
2. abecrpcpass= [rpcpass in abec.conf]
```

In addition, in order to let **abectl** (abewallet control console) interact with abewallet later, we need to set the following options in abewallet.conf.

```
1. rpcuser= [whatever_username]
2. rpcpass= [whatever_password]
```

**Suggestion:** It can be confusing to get the correct rpc username and password for inter-process communication. To simplify the username and password management, we may consider setting the **rpcuser** and **rpcpass** in abewallet.conf to be the same as the **abecrpcuser** and **abecrpcpass**, respectively.

Besides the configuration above, you need to do a few more configurations as follows if you are running abec and abewallet at different machines.

## 4.1 abec and abewallet are running on different machines

Add the following option in **abec.conf**:

```
1. rpclisten=[IP:PORT]
```

or simply:

```
1. rpclisten=
```

so that your **abec** will listen to all interfaces on the default port number (i.e. 8667).

Next, add the following option in **abewallet.conf** for specifying the IP address and PORT of the remote full node (**abec**).

```
1. rpcconnect=[IP:PORT]
```

IP and PORT are the IP address and listening port of abec, respectively. If a port is not specified, the default port (8667) will be used.

Go to the configuration folder of **abec**, **delete** the two rpc files, namely *rpc.cert* and *rpc.key*, and run the **abec** again for generating a new pair of *rpc.cert/rpc.key* in the configuration folder of **abec**.

Finally, copy `rpc.cert` from the configuration folder of **abec**, and store the file as `~/Documents/abel/rpc.cert` and set the `cafile` option in the local `abewallet.conf` as

```
1. cafile=~/abel/rpc.cert
```

## 5. Run a Full Node - abec

---

Go to `~/abel/abec-linux-amd64-v0.11.9/` and run a full node (**abec**):

```
1. # macOS and Linux
2. ./start_abec.sh
3. # Windows
4. abec
```

## 6. Run a Wallet - abewallet

---

Go to `~/abel/abewallet-linux-amd64-v0.11.9/` and run the wallet (**abewallet**):

```
1. # macOS and Linux
2. ./start_abewallet.sh --walletpass=[your public passphrase]
3. # Windows
4. abewallet --walletpass=[your public passphrase]
```

## 7. Use abectl to Operate on abec and abewallet

---

In the **abec** folder, there is another executable called **start\_abectl.sh** (macOS and Linux) or **abectl** (Windows). This executable can be used for checking the balance of your wallet, make fund transfer, and so on.

### 7.1 Check balance

Run the following to check the balance of your wallet where **username** and **password** should be the same as the options **rpcuser** and **rpcpass** we configured in **abewallet.conf**, respectively.

```
1. # macOS and Linux
2. ./start_abectl.sh --rpcuser=[username] --rpcpass=[password] --wallet getbalancesabe
3. # Windows
4. abectl --rpcuser=[username] --rpcpass=[password] --wallet getbalancesabe
```

## 7.2 Unlock

Before sending a transaction or generating a new address, you need to unlock your wallet first.

1. *# macOS and Linux*
2. `./start_abectl.sh --rpcuser=[rpc username] --rpcpass=[rpc password] --wallet walletunlock [private passphrase] [timeout]`
3. *# Windows*
4. `abectl --rpcuser=[rpc username] --rpcpass=[rpc password] --wallet walletunlock [private passphrase] [timeout]`

The unit of timeout is in seconds.

Example:

1. `./start_abectl.sh --rpcuser=abewalletrpcuser --rpcpass=abewalletrpcpass --wallet walletunlock 123456 240`

This unlocks the wallet with passphrase 123456 for 240 seconds.

## 7.3 Generate a new address

To generate a new wallet address, unlock the wallet and execute the following.

1. *# macOS and Linux*
2. `./start_abectl.sh --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet generateaddressabe`
3. *# Windows*
4. `abectl --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet generateaddressabe`

## 7.4 Query the Maximum Number of Addresses

Everytime when the wallet is recovered or a new transaction is initiated, there is a fresh wallet address is created. We can run the following command to find out the total number of addresses create:

1. *# macOS and Linux*
2. `./start_abectl.sh --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet addressmaxsequencenumber`
3. *# Windows*
4. `abectl --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet addressmaxsequencenumber`

## 7.5 Make a transfer

As an Abelian wallet address is long, we find it cumbersome to paste a wallet address directly into the command line. Instead, we can create a file called **arg1** in the configuration folder of abec and add receiver's address and amount into it. The format is as follows.

```
1. [  
2.   {  
3.     "address":"addr1",  
4.     "amount":700000000  
5.   },  
6.   {  
7.     "address":"addr2",  
8.     "amount":500000000  
9.   }  
10.]
```

Note that the unit of amount is **Neutrino** (1 ABE = 10,000,000 Neutrino).

And then we can send transaction by

```
1. # macOS and Linux  
2. ./start_abectl.sh --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet sendtoaddressesabe -  
3. # Windows  
4. abectl --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet sendtoaddressesabe -
```

Example:

```
1. $ cat /Users/username/Library/Application Support/Abec/arg1  
2. [{"address":"addr1", "amount":700000000}, {"address":"addr2", "amount":500000000 }]  
3. $ ./start_abectl.sh --rpcuser=[rpcuser] --rpcpass=[rpcpass] --wallet sendtoaddressesabe -
```

We omit the rest of the address for simplicity. This means we send 70 ABEL to addr1 and 50 ABEL to addr2.

## 8. Recover a wallet

### 8.1 Prepare the information

If you want to recover a wallet, *please first prepare the information as described below:*

1. *Please make sure your wallet mnemonics are available.*
2. *Query the maximum number of addresses in the current wallet.*
3. *Query the balance information of the current wallet.*
4. *Backup the abewallet configuration folder.* Please copy the folder to somewhere else which is also safe. This, together with the public and private passphrase, may help rollback to the current wallet in case your mnemonics are not as you expected.

## 8.2 Recover the wallet

After finishing the preparation, delete `logs/` and `mainnet/` folders in the configuration folder of **abewallet**, and run

1. *# macOS and Linux*
2. `./start_abewallet.sh --create`
3. *# Windows*
4. `abewallet --create`

Example:

1. `$ ./start_abewallet.sh --create`
2. **Enter** the **private** passphrase **for** your **new** wallet:
3. **Confirm** passphrase:
4. **Enter** the **public** passphrase **for** your **new** wallet:
5. **Confirm** passphrase:
6. NOTE: **Use** the `--walletpass` option to configure your **public** passphrase.
7. **Do** you have an existing wallet seed you want to **use**? (n/**no**/y/**yes**) [**no**]: y
8. **Enter** the crypto version **is:0**
9. **Enter** existing wallet mnemonic: biology,hazard,sudden,dignity,drop,jealous,butter,believe,answer,enter,practice,scorpion,health,tunnel,rival,vault,neutral,season,proof,must,path,steel,**final**,female
10. **Please** input the max **No.** of address to recover :5
11. **Creating** the wallet...
12. 2022-05-07 19:55:01.548 [INF] WLLT: **The** addresses **with No. in** [0, 5] have been restored.
13. 2022-05-07 19:55:02.282 [INF] WLLT: **Opened** wallet
14. **The** wallet has been created successfully.

**Note:** For the `max No. of address to recover` here we should input **the number obtained in the preparation in 8.1.**

## 8.3 Check the wallet

After recovering the wallet, to be safe, please check whether it is recovered as expected:

1. Configure the wallet and connect it to an abec node, so that it can synchronize the Abelian blockchain and find all the ABEL coins.
2. After the synchronization, query the balance and check whether it is as expected.

## 9. Upgrade to the Latest Version

If you already have an old version running (say v0.11.1), you can follow the steps below to upgrade to the latest version (v0.11.9 as of this writing).

First, **stop** running both **abec** and **abewallet** of the old version.



Next, **download** and **unzip** the two compressed files for **abec** and **abewallet**.

To run the new **abec**, simply go to the new **abec** folder and run it.

## 9.1 Run the new version of abewallet

This part is bit tricky as we need to remove the old configuration folder of **abewallet** first, then recover your wallet before running the new version.

### 9.2.1 Prepare the information

Please first prepare the information as described in 8.1.

### 9.2.2 Upgrade the wallet

First, **remove** the abewallet configuration folder:

- Windows: C:\Users\username\AppData\Local\Abewallet
- macOS: /Users/username/Library/Application Support/Abewallet
- Linux: /home/username/.abewallet

Next, go to the new **abewallet** folder and run:

```
1. # macOS and Linux
2. ./start_abewallet.sh --create
3. # Windows
4. abewallet --create
```

Press control+C. By doing this, we re-create the configuration folder.

Then, do the configuration of **abewallet** as described in Section 4.

After that, go to the new **abewallet** folder and recover your wallet:

```
1. # macOS and Linux
2. ./start_abewallet.sh --create
3. # Windows
4. abewallet --create
```

Example:

```
1. $ ./start_abewallet.sh --create
2. Enter the private passphrase for your new wallet:
3. Confirm passphrase:
4. Enter the public passphrase for your new wallet:
5. Confirm passphrase:
6. NOTE: Use the --walletpass option to configure your public passphrase.
7. Do you have an existing wallet seed you want to use? (n/no/y/yes) [no]: y
```

8. **Enter** the crypto version **is:0**
9. **Enter** existing wallet mnemonic: biology,hazard,sudden,dignity,drop,jealous,butter,believe,answer,enter,practice,scorpion,health,tunnel,rival,vault,neutral,season,proof,must,path,steel,**final**,female
10. **Please** input the max **No.** of address to recover :5
11. **Creating** the wallet...
12. 2022-05-07 19:55:01.548 [INF] WLLT: **The** addresses **with No. in** [0, 5] have been restored.
13. 2022-05-07 19:55:02.282 [INF] WLLT: **Opened** wallet
14. **The** wallet has been created successfully.

**Note:** For the  here we should input **the number obtained in the preparation in 9.2.1.**

Finally, run the wallet (**abewallet**):

1. *# macOS and Linux*
2. `./start_abewallet.sh --walletpass=[your public passphrase]`
3. *# Windows*
4. `abewallet --walletpass=[your public passphrase]`