User's Guide

1 Current release
   1.1 What is Beam? ................................................................. 3
   1.2 Getting Started .............................................................. 4
   1.3 Important differences from other cryptocurrencies .................. 4
   1.4 Reporting Issues and Getting Support .................................. 5
Note: Beam is currently in Mainnet.

Rules Signature: 6d622e615cfd29d0 (for height >= 321321)

Download binaries from: Beam Downloads Page
Source code: Beam Github
Double Doppler 4.0
This release includes desktop wallet support for Atomic Swaps, sync time improvements and much more...

Known limitations:
1. Inconsistent transaction statuses when running two or more wallets that use the same wallet.db file, cloned manually.
2. If multiple wallets are restored from the same seed phrase, transaction history and addresses will not be shared among the wallets.

Points to mention:
In case a mobile wallet is connected to a random node and the user has both mobile and a desktop wallets which are using the same seed phrase, the funds sent to the desktop wallet won’t appear on the mobile wallet. If the desktop wallet is running a local node, it will see funds sent to both wallets. The reason is that the local node (integrated into desktop wallet) always monitors the blockchain for UTXOs related to the seed of the wallet. The mobile wallet does not run a local node and thus can only monitor transactions sent to its specific SBBS addresses. Of course, no funds will be lost in any event.

TL;DR
I want to Mine Beam
I want to learn how to use Beam Desktop Wallet
I want to learn how to use Beam Command Line Wallet

1.1 What is Beam?

Beam is a next generation scalable, confidential cryptocurrency based on an elegant and innovative Mimblewimble protocol.
Things that make BEAM special include:
Users have complete control over privacy - a user decides which information will be available and to which parties, having complete control over his personal data in accordance to his will and applicable laws.

Confidentiality without penalty - in BEAM confidential transactions do not cause bloating of the blockchain, avoiding excessive computational overhead or penalty on performance or scalability while completely concealing the transaction value.

No trusted setup required.

Blocks are mined using Equihash Proof-of-Work algorithm.

Limited emission using periodic halving.

No addresses are stored in the blockchain - no information whatsoever about either the sender or the receiver of a transaction is stored in the blockchain.

Superior scalability through compact blockchain size using the “cut-through” feature of Mimblewimble.

No premine. No ICO. Backed by a treasury, emitted from every block during the first five years.

Implemented from scratch in C++.

1.2 Getting Started

The simplest way to get started with Beam is by visiting the Beam website, reading and understanding the materials posted there and joining Beam Community on Telegram for updates and discussions.

**Danger:** Beam is extremely new and experimental technology. No guarantees can be provided by anyone whatsoever. Use it at your own risk. Make sure you know what you are doing, especially if there is money involved.

Just like any other cryptocurrency, using Beam requires learning and understanding what this all is about. If real money is involved, it also requires concern with security of the process.

**Hint:** That said, you can always safely play with Beam by connecting to the permanent Testnet.

To learn more about how cryptocurrencies work in general and Beam in particular please visit our Resources page. Once you familiarized yourself with key ideas and concepts, it is recommended to start from connecting to our Testnet. The simplest way to do that is by downloading and installing our Desktop Wallet and following instructions in Desktop Wallet User Guide.

1.3 Important differences from other cryptocurrencies

Mimblewimble has several important differences from most other existing cryptocurrencies which are very important to understand. Please review the following information carefully.

**The concept of Address is completely different**

In most cryptocurrencies Address is a hashed public key for which the owner of that Address knows the corresponding private key. In order to transfer funds, the Sender should only know the Address of the Receiver in order to create a unilateral transaction. *The Sender is not aware of whether the Receiver is online or not or whether it even exists.* Once transaction to an Address is complete and added to the blockchain, Receiver that can prove knowledge of the private key corresponding the Address can control this UTXO (short for Unspent Transaction Output).
In Mimblewimble there are no addresses at all and transaction are created interactively by both Sender and Receiver wallets. This means that in order to create a transaction, both wallets have to participate in the creation process and eventually co-sign the transaction before it is sent to the blockchain.

**Attention:** In Beam it is not possible to create a transaction unilaterally. Both Sender and Receiver have participate in transaction creation.

To allow Sender and Receiver wallets to create transactions without having to be online at the same time and directly connected to each other, Beam added a module called **SBBS** that allows wallets to securely communicate using encrypted messages to create a transaction. SBBS Addresses are merely private / public key pairs used to encrypt and decrypt these messages.

**Important:** SBBS Addresses are not recorded in the blockchain and are not used to control funds

You are encouraged to create a new SBBS Address for each transaction.

**Wallet and Node concepts are slightly different**

Beam documentation mentions terms Wallet and Node quite a lot and it sometimes causes confusion with users of other cryptocurrencies.

Beam Wallet is a light client which stores information about UTXO that belong to it and has an ability to create new transactions by connecting to other wallets via **SBBS**. It does not store or verify the entire blockchain and can thus only work if connected to a Node.

Beam Node, is a full node that downloads, validates and updates the entire blockchain state.

**Note:** Beam Desktop Wallet, provides options to run both as just the Wallet (connected to a remote node) and as a full node.

**Attention:** It is always recommended to run a full node

**Information that can be restored from the blockchain is completely different**

In most blockchains, information about current UTXOs and the transaction history can be recovered from the blockchain using only the Seed Phrase.

In Beam, only UTXOs can be recovered from the blockchain. All other information, including transaction history and any other meta data are only stored locally in the Beam Wallet database and encrypted by Wallet Password.

This means that if you run Beam Wallet on two different machines, transaction history WILL NOT be synchronized. This also means, that to preserve transaction history, or any additional meta data, it is necessary to regularly backup Beam Wallet database file.

For more information about backup and restore procedure see Backup and Restore

**1.4 Reporting Issues and Getting Support**

To report issues and get support please perform the following steps:

1. Gather all relevant information including:
• Detailed description of the problem you have encountered and steps to reproduce it
• Version of the binaries you are running
• Logs (see Log locations for information where to find the logs files)
• Relevant configuration files (please check for private information before sending)
• Your system configuration
• Screen shots or any additional information you think is relevant

2. Send an email to support@beam.mw (or testnet@beam.mw for testnet related issues).

You can also open an issue in github and follow the provided template.

Attention: Providing all the information described above will allow us to quickly and efficiently analyze and resolve the issue for you and everyone else.

Warning: The following document is still under construction and is subject to changes.

1.4.1 Desktop Wallet User Guide

Beam Desktop Wallet is the simplest way to start using Beam. It is available for Linux, Mac and Windows platforms (see Supported Platforms for details).

Attention: Beam blockchain does not store transaction history and SBBS addresses. These are only stored in local database inside the wallet data folder.

Please follow the guidelines below to avoid problems with sending are receiving Beam transactions.

1. Do not copy the wallet.db to another machine and run another wallet simultaneously using the same wallet database
2. Do not run two different wallets with the same seed at the same time
3. SBBS messages sent between wallets expire after 12 hours. You have to connect within 12 hours of the transaction initiation to receive or send the funds.
4. SBBS Addresses by default expire after 24 hours. Always use ‘never’ expiring addresses with pools and exchanges to make sure you receive payments.

Downloading binaries

Start downloading here: https://www.beam.mw/downloads/mainnet

Ensure that the communication between your browser and Beam’s official website is encrypted by verifying that the padlock icon is displayed in its locked state on the URL bar.

Download the Beam wallet app for your platform (Mac, Linux or Windows):

• On Mac open terminal and run: shasum -a 256 <file>
• On Linux run: `sha256sum <file>`

• On Windows, open Power Shell and run: `Get-FileHash 'C:\Users\<your_user_name>\Downloads\beam-node-1.3860.zip`

Substitute your own path instead of the one in the example above.

**Note:** As the wallet will be continuously updated, the actual version numbers and SHA values might be different from the screenshot at the moment of reading.

## Installing the desktop app

Once the application image is downloaded, double-click the icon to start the installation.

### Mac

When you click on the .dmg file you will see a screen with Disclaimer and End User License Agreement. Please read carefully and click ‘Agree’.

On the next screen, drag the Beam Wallet icon into the Applications folder to install.

When you will try to open the wallet for the first time, you will receive a security warning stating that Beam Wallet was created by an unrecognized developer. We are working to eliminate the warning, meanwhile follow the steps below to launch Beam Wallet app on your Mac.

Open ‘System Preferences’.

Locate and click ‘Security & Privacy’ settings.
Beam Wallet:

Short Summary

PLEASE NOTE: Beam is free open-source software. While we do our best to build the most secure software we can, but we do not hold any responsibility if anything goes wrong. Your use of Beam Wallet and Beam Blockchain is at your own risk. You may lose your money, your software may become unusable, your computer may crash and other bad things may happen. The software is licensed under Apache 2.0 license and your use in the software is subject to the terms mentioned below.

Terms of Use

Please read this terms of use agreement (the "Terms") carefully as they form a binding legal agreement between you and Beam Development Limited ("Beam", "us", "we", "our"), that governs the use of the digital wallet developed by Beam that allows to store, track, transfer and manage virtual currency (the "Software" or "Beam Wallet"). The term "you" refers to the person accessing or using the Software, or the entity or organisation on whose behalf such person access our Software.

1. Risks Related to the Use of the Software. Beam is not be responsible for any losses, damages or claims arising from events falling within the scope of the following: (i) mistakes made by you of any Software, forgotten passwords, payments sent to wrong Wallet Address (as defined below), and accidental deletion of Beam Wallets; (ii) software problems, e.g., corrupted Beam Wallet files, improperly constructed transaction, proof of stake problems.
1.4. Reporting Issues and Getting Support
A login password has been set for this user. 

- **Require password immediately** after sleep or screen saver begins
- Show a message when the screen is locked
- Disable automatic login

Allow apps downloaded from:
- **App Store**
- **App Store and identified developers**

Click the lock to make changes.
Click the lock icon in the bottom left corner of the dialog to unlock.

![Security & Privacy settings](image)

Change the ‘Allow apps downloaded from’ setting from ‘App Store’ to ‘App Store and identified developers’. Click the lock icon again when done making changes.

Now, launch the Beam Wallet from the Applications folder. When notification appears, click ‘Open’.

**Where are the files?**

Once Beam Wallet desktop app is installed, the wallet data files are stored separately from the binaries.

The locations of all the files are described here: *Files and Locations*

**Note:** When the Beam wallet app is running, right click on it and select *Options | Keep in dock* for easy access of the app in the future.
A login password has been set for this user. Change Password...

- Require password immediately after sleep or screen saver begins
- Show a message when the screen is locked
- Disable automatic login

Allow apps downloaded from:
- App Store
- App Store and identified developers

Click the lock to make changes.

"Beam Wallet" is an app downloaded from the Internet. Are you sure you want to open it?
Chrome downloaded this file yesterday at 12:44 from www.beam.mw.

Cancel  Show Web Page  Open
Creating new wallet

Once you launch the wallet for the first time, click ‘Create new wallet’

Generating seed phrase

As a part of creating a new wallet, a new seed phrase will be generated for you.

**Attention:** Seed phrase is the most important secret you have to keep. Knowing the seed phrase enables you (or anyone else) to access all your funds.

**Attention:** Seed phrase in the Beam wallet is not linked to email, phone number or any other identifier. You will need this phrase to restore your wallet when you lose or reformat your device, or want to access your funds from another device (your mobile phone or another desktop / laptop).

**Attention:** The seed phrase is for your eyes only! Make sure no one is looking over your shoulder. For the best security always generate it on a clean air-gapped machine.

Did you wrote down your seed phrase correctly? Triple-check your handwriting again. The difference between _F_unnel and _T_unnel can be crucial when trying to restore a wallet with valuable funds in the far or near future.

Did you verified your handwriting? Now go find a safe space for the paper!

**Important:** Storing the seed phrase on your computer makes your funds prone to cyber attacks (read: much less secure). 'Creative’ approaches like saving a screenshot of the wallet or your handwriting on your computer or in the cloud may sound like a good idea, but it is absolutely not recommended. If hackers get the access to your computer, network drive etc., they can can potentially steal your seed phrase by using OCR programs (which can scan pictures...
Create new wallet

Create new wallet by generating seed phrase. If you ever lose your device, you will need this phrase to recover your wallet. Never type your seed phrase in keychains or password managers. Never save it in your local or remote folders in any form.

- Do not let anyone see your seed phrase
- Never type your seed phrase into password managers or elsewhere
- Keep the copies of your seed phrase in a safe place
Create new wallet

Your seed phrase is the access key to all the cryptocurrencies in your wallet. Print or write down the phrase to keep it in a safe or in a locked vault. Without the phrase you will not be able to recover your money.

1 garden  2 water  3 rifle  4 century
5 mutual  6 foster  7 wear  8 fantasy
9 deer  10 attend  11 approve  12 maple

← back  → next
and transform them into plain text) and, therefore, get access to your funds.

**Attention:** Always store your seed phrase in a safe and secure location (and better more than one in different geo locations). Write it on a piece of paper. Do not store electronically neither as plain text nor in any other form!

In order to ensure that you have really written down your seed phrase, you will be asked to fill in the specific words from your seed phrase in random order.

Only when you typed all the selected words correctly, you will be allowed to proceed to the next step.

**Setting wallet password**

To access your Beam Wallet, you will need to create a password. This password is not the same as the seed phrase. Seed phrase identifies a wallet and enables access to all the funds stored in it from any device. Your wallet password provides with a second security layer in case someone gains access to your computer or has stolen your wallet database file. It is important to choose a strong password.

The wallet will provide some indication of password strength for your convenience. Do not count on it, however. Choose a password that is at least 8 characters long with a combination of letters, numbers, and symbols.
Create new wallet

To ensure the seed phrase is written down, please fill-in the specific words below:

1.  
2.  
3.  
4.  
5.  
6.  
7.  
8.  
9.  
10.

← back  → next
Create new wallet

To ensure the seed phrase is written down, please fill-in the specific words below

1. rifle
2. wear
3. water
4. maple
5. century
6. foster
Create new wallet

Create password to access your wallet

Enter password

***

Very weak password

Confirm password

← back  → next
Choosing the node connectivity mode

Beam Desktop Wallet can connect to the network through:

- Integrated node
- Random remote node
- Specific remote node

If you choose to run an integrated node from within the wallet, the trusted node will automatically verify the blockchain. This means you will automatically be connected to a node when you open your wallet.

**Note:** When running behind a firewall you can change the default port the node will be listening on. You will be probably provided at least one default peer to connect to but you can always add more peers on the Settings screen. The recommended peers are published in the list of bootstrap nodes in the downloads page on Beam official website.

Random mode allows you to automatically connect to random bootstrap node. In this mode Beam Wallet acts like a ‘light client’, it will create transactions but will have to trust the remote node for blockchain verification. It is recommended for lighter devices with limited CPU power and/or RAM memory.

If you are running your own node somewhere (or maybe even more than one) and want to connect specifically to it, use the third option by providing the IP and port the node is listening on.
Setup node connectivity (testnet)

- Run integrated node (recommended)
- Connect to random remote node
- Connect to specific remote node
Setup node connectivity (testnet)

- Run integrated node (recommended)
- Connect to random remote node
- Connect to specific remote node
  127.0.0.1:10000

← back  → proceed to your wallet
Synchronizing the wallet

Once the wallet is connected, it synchronizes with the current blockchain state from the network. Be patient when running with the integrated node: the data downloading process might take some time. The wallet will first download and validate the latest Macroblock and then all the rest of the blockchain.

Upgrading from previous version

If during installation the wallet discovers one or more previous versions installed it will propose to migrate to a new version.

If you do not wish to migrate an existing wallet, you can also create a new wallet or choose the location of the wallet database file yourself by clicking ‘Login to another wallet or create a new one’.

Note: Migration will copy the existing wallet database (wallet.db) keeping all the data intact.

In the next screen you will be asked to choose which wallet database file you want to migrate. If the file you need does not appear in the list you can select it manually using ‘Select file manually’ option.

Once file is selected, click ‘next’ to enter database password.

Once you enter the correct password, the migration will complete and the wallet will resume synchronization with the blockchain.
Scalable confidential cryptocurrency

Your wallet will be migrated to v1.0.4185

Start migration

Login to another wallet or create new one
Select the wallet database file

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>...ton Support/Beam Wallet/1.0.3978/wallet.db</td>
<td>648 kb</td>
<td>1/17/19 10:00 AM</td>
</tr>
</tbody>
</table>

Login to another wallet or create new one
Main screen

Once your wallet is created, the main screen will show up. In the future, the screen will pop-up automatically after you open the application and type in your password.

Wallet status indicator

The main screen of the wallet shows the current balance as well as the transaction history and statuses. On the left, under the Beam logo, there is a toolbar that provides navigation between different wallet screens such as Main Screen, Addresses Screen, UTXO Screen and Settings Screen.

In the top left corner of the Main Screen, under ‘Wallet’, you see connection status indicator.

Green indicator means that the node is up and running connected to peers.

Red means it is either unable to connect to the node or the node has a problem you can help to solve.

In brackets, the indicator also specifies the network to which the wallet is connected:

- **Mainnet**: the network with real money and actual transactions.
- **Testnet<number>** (such as Testnet3, Testnet4 etc) : staging environments for trying new features in our wallet, node and miner software. For advanced users only.
- **Masternet**: new features under development, if you see this name in your wallet it means that you are very early adopter or Beam code contributor otherwise you've probably arrived to the wrong place.
Correct:
- online

Wrong:
- online (testnet 4)

Error:
- wallet is not connected to the node
Attention: Since Mainnet is the default network for the vast majority of Beam users, nothing is written by the online status.

Finally, node connectivity node is displayed (ie. integrated, random remote node or specific node).

Financial transactions

‘Send’ and ‘Receive’ buttons at the top right corner help us Sending BEAM and Receiving BEAM.

Main screen password

In the future, the screen will pop-up automatically after you open the application and type in your password.

Receiving BEAM

Before starting to receive BEAM for the first time, please read first about what Address is.

Here is how the process of receiving BEAM looks like from a Receiver’s perspective:

- Generate an address
- Send your address to the Sender person over a secure communication channel
  - Both Sender and Receiver’s Wallet must be online at the same time to complete a transaction.

It’s possible to reuse an address that already exists, more on that later.

Generate an address

Proceed to the main screen and click the blue ‘Receive’ button at the top right corner. This will open the receive screen.

Copy and paste the newly generated Beam address to send to Sender over a secure communication channel. There are three ways to do it:

- By selecting the address and clicking Command-C or Ctrl-C (depending on your platform)
- By right-click on the address and choosing ‘Copy’ from the drop-down menu
- By clicking the ‘Copy’ button

Each time the Receive Beam dialog is open, a new Beam address is generated. By default, the address is valid for 24 hours. It is best to give the Sender your address closest to the time they will be sending BEAM so the address does not expire.

You can explicitly set the expiration time to ‘Never’ for this address only by selecting the value in the ‘Expires’ drop down.

Warning: The permanent addresses that never expire are only useful in limited special cases, like working with mining pools or exchanges. For ultimate privacy, do not use this option for regular transactions, rather always create a new address for each transaction.
1.4. Reporting Issues and Getting Support
Attention: You have to click ‘Close’ button in the screen for the address to become active.

QR code

If the Sender uses a mobile app, he can quickly scan the QR code instead of receiving, copying and pasting the alphanumeric address. This feature will be available soon in the upcoming Beam Wallet mobile app.

Comment

You can add a comment when creating the receiving address. The comment is never sent to the network, it is only visible inside your wallet and is used for internal bookkeeping only.

The comment can be seen on the Address screen and in the extended transaction view.

Sending the address

Attention: When sending the address make sure you use a secure communication channel.
Attention: Make sure the entire address is sent to the Sender as it’s longer than it appears on the screen. Don’t forget to double check the value in whichever messenger app of your choice because viruses and malware on your computer may change your address while it’s in the clipboard.

Completing the transaction

Once Sender initiates the transaction, you will see on the new transaction appear in the transaction list on the main screen. The amount sent will also appear on the ‘In progress’ box.

Normally, a transaction will pass through the following stages:

- **In Progress** - the phase when the transaction is being created by the Sending and Receiving wallets.
- **Confirming** - the phase after the transaction was sent to the nodes, but before it is mined.
- **Completed** - the phase after the transaction is fully mined and confirmed.

Once transaction is complete, the available balance will be updated and the ‘In Progress’ box will disappear.

Sending BEAM

Before starting to send BEAM, please read first about what *Address* is.

Here is how the process of sending BEAM looks like from a Sender’s perspective:
• Receive the address the funds should be sent to
• Send BEAM to Receiver
• Stay online until Receiver confirms the transaction

Receiving the address

**Attention:** Make sure that the address is received untampered by using a secure communication channel.

**Attention:** When copying the address to the Beam Wallet app please verify visually that the address in the wallet looks exactly like the address in the secure messaging app, because viruses and malware on your computer may change your address while it’s in the clipboard.

Sending funds

In order to send BEAM, you will need to click the magenta ‘Send’ button at the top right corner. This will open the Send screen.

Make sure you have the correct address and paste the Receiver’s Beam address in the ‘Send To’ field.
To help to identify the transaction, you may also choose to fill in the optional Comment field. The comment will remind you what or who the transaction is for. The comment is stored locally, thus it will only be visible in your wallet for bookkeeping purposes.

The comment can be seen on the Address screen:

![Address screen](image)

The comment is also displayed in the extended transaction view on Main Screen:

Select the transaction amount in BEAM you want to send. Transaction amount is in BEAM and may contain fractional values such as 1.25 BEAM or 11.3 BEAM and the like. Keep in mind you also have to pay a transaction fee, hence the amount to send plus the fee must be equal to or less than the available balance.

Transaction fees are specified in GROTH (100 millionths of BEAM). Amount of fees you need to pay depends on the current status of the network and average fee sizes. Simply said, the higher transaction fee will help miners to prioritize your transaction. To determine the current average fee size use Beam Blockchain Explorer.

You can see the remaining amount of BEAM in your wallet and the change that will be received after the transaction. After you click ‘Send’ you will see a confirmation with the most important transaction details:

**Completing the transaction**

Once you confirm, the transaction is sent to the Receiver’s wallet. If Receiver’s wallet is currently offline or if the network is loaded, you might see the transaction appear ‘In Progress’ on your transaction list. Once the transaction is complete, it will be sent to the nodes and shown as ‘Confirming’.
1.4. Reporting Issues and Getting Support
1.4. Reporting Issues and Getting Support
Note: While a transaction is in ‘In Progress’ you can cancel it by clicking on the dropdown to the right of the transaction row and then select ‘Cancel’. The other party will receive notification that the transaction was either ‘Canceled’ or ‘Failed,’ and funds plus fee that were allocated for this transaction will become available again. It is not possible to cancel a transaction in ‘Confirming’ or ‘Completed’ states.

Warning: If your transaction appears as ‘In Progress’ for a long time, it means the Receiver is not online.

Attention: If the transaction was not sent to the nodes, for any reason, it will expire after 1440 blocks, or roughly 24 hours. This is done to avoid a situation in which UTXO is locked forever.

Restoring funds

This process allows you to restore your funds directly from the blockchain. It is useful in the scenarios such as:

- You’ve got a new device and would like to use your wallet on it
- You forgot your local password and can’t access your funds
Attention: With Beam, only funds are stored on the blockchain. Everything else, such as your active addresses, contacts or transaction history can’t be restored.

Note: Very soon exporting transactions history for backup and bookkeeping purposes will be implemented.

Before restoring funds on a machine the wallet was already installed, manually remove the wallet database file as described in Files and Locations. No action is required on a new machine.

Start the BEAM desktop wallet app and press ‘Restore wallet’ button.

You will be asked to enter your seed phrase. Time to get the phrase out from your safe locker and type the words in.

Warning: If a wrong word was typed or an existing word was misspelled, your funds will not be restored successfully. Example: ‘litt_el_’ instead of ‘lit_le_.’

Did you checked your spelling? Once you are sure, click ‘Restore wallet.’

Upon completion, you’ll see the main screen of the wallet with your restored funds.
Restore wallet

Type in your seed phrase

[Input fields for seed phrase]

← back  ▶️ restore wallet
1.4. Reporting Issues and Getting Support

Restore wallet

Type in or paste your seed phrase

1. garden 2. water 3. rifle 4. century
5. mutual 6. foster 7. wear 8. fantasy
9. deer 10. attend 11. approve 12. maple

← back ↻ restore wallet
Beam Documentation, Release 2019

Chapter 1. Current release

Scaleable confidential cryptocurrency

Restoring wallet...
Scanning UTXO 25/210203, found 137.078 BEAM
(estimated time left: 1h 5 min)

Only the wallet balance (UTXO) can be restored.
Transaction info is always private and never kept in the blockchain.
Note: Please be patient, restoring funds is a thorough and time consuming operation.

Attention: If the available balance is zero, it means that one or more words from your seed phrase weren’t typed correctly or are wrong.

Address

Let’s define the meaning of address in the BEAM ecosystem: BEAM is always sent from one address to another. Both sending and receiving addresses are alphanumeric tokens that uniquely define the transaction endpoints.

A person can create as many addresses as required. The address creation process is explained in Receiving BEAM.

Example: There are two wallets: one belongs to you and the other belongs to Alice. You have created one address to receive money from Alice and another address to send money to Alice. Since you can generate multiple addresses, Alice will never know that she’s receiving money from the same person that she is sending the money to (unless you want her to know).

Example: There are three wallets: one belongs to you, one belongs to Alice and another belongs to Bob. You have created one address to receive money from Alice and another address to receive money from Bob. Alice and Bob will never know that they are sending money to the same person.

Same address can be used for sending and receiving money.

Example: you have created an address to send money to Alice. Alice can see the address the money came from and can send money to the address back to you.

Attention: For ultimate privacy, it is advised to have a dedicated address for every transaction (ie. for both Sending or Receiving).

Attention: Although not recommended, an address can be reused until it had reached its expiration (24 hours since when it was generated).

Example: Imagine you’ve created the address with expiration interval of 24 hours and immediately sent it to Alice. In the next 24 hours, Alice will be able to send BEAM to you as many times as she likes, reusing the same address of yours.

Warning: Reusing same addresses, created with longer expiration interval, can be convenient yet the tradeoff of decreased privacy should be kept in mind.

Address screen

The screen lists all the addresses that were used in all incoming and outgoing transactions. All the data in this screen is only stored locally in your wallet and is not related to the blockchain in any way.
Note: For your own privacy, a new address is generated for each outgoing transaction. Yes, you heard that right! That address can be seen in your active or expired addresses list.

Upon Beam Wallet desktop app installation, a single address is created by default. The address has a default expiration time of 24 hours.

You can always create a new one by using Receive Screen. You can see all your active addresses in the ‘My Active Addresses’ tab.

[TODO: screenshot]

Once an expired address becomes inactive, it cannot be used in any incoming or outgoing transaction. Each address has a default expiration time of 24 hours, including the default address. An expired address becomes inactive and you won’t be able to be use it again. You can see the expired addresses listed under ‘My Expired Addresses.’

When you’ll click the three dots located to the right of any address, the menu with additional address actions will open. You can manually edit or delete any address choosing ‘Edit’ or ‘Delete’ from the menu.

You can manually expire any active address by choosing ‘Expire address’ from the drop down menu.

[TODO: screenshot with address edit dialog]

For privacy concerns, you can manually delete any address by clicking the three dots located to the right of the address and choosing ‘Delete address’ from the drop down menu.

[TODO: screenshot with delete address]
### Addresses

- **Comment**: Default
- **Address**: iRqek...fign7h
- **Category**: 
- **Expiration Date**: 1 Jan 2019 | 2:01 PM
- **Created**: 31 Dec 2018 | 2:01 PM

---

1.4. Reporting Issues and Getting Support
In the ‘Contacts’ tab, you can see every address that sent you BEAM or to which you’ve sent BEAM to.

**UTXO**

UTXO (Unspent Transaction (TX) Output) is like a banknote of a specific amount. Simply said, if BEAM is the currency, any UTXO can be considered a ‘bill’. You can have multiple ‘bills’ in your wallet at the same time.

**UTXO screen**

On the technical level, in Beam, like in most other cryptocurrencies, your balance emerges as a result of multiple incoming and outgoing transactions. Each transaction uses some existing inputs and creates new outputs. All the outputs controlled by the wallet are shown in the UTXO screen.

The type of UTXO can be:

- **Regular** - UTXO received as a result of a transaction. It is immediately available for spending
- **Change** - UTXO received as a result of change from a transaction. It is immediately available for spending
- **Transaction fee** - Fees received as a result of mining a block which contain transactions
- **Coinbase** - UTXO you have mined. It has maturity of 3 hours (240 blocks) and will not be immediately seen in Available tab
## Chapter 1. Current release

### UTXO

<table>
<thead>
<tr>
<th>Amount</th>
<th>Height</th>
<th>Maturity</th>
<th>Status</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.99999 B</td>
<td>35000</td>
<td>35002</td>
<td>available</td>
<td>Change</td>
</tr>
<tr>
<td>3.99999 B</td>
<td>34929</td>
<td>34931</td>
<td>available</td>
<td>Change</td>
</tr>
<tr>
<td>9 B</td>
<td>30718</td>
<td>30690</td>
<td>spent</td>
<td>Regular</td>
</tr>
<tr>
<td>40 B</td>
<td>30688</td>
<td>30690</td>
<td>available</td>
<td>Regular</td>
</tr>
<tr>
<td>5 B</td>
<td>30688</td>
<td>30690</td>
<td>spent</td>
<td>Regular</td>
</tr>
<tr>
<td>36 B</td>
<td>30683</td>
<td>30685</td>
<td>spent</td>
<td>Regular</td>
</tr>
</tbody>
</table>
UTXO details

Click on the specific UTXO you want to see more details of.
{TODO: screenshot}

UTXO in sending BEAM screen (explained by example)

Assume that you have 100 BEAM in a single ‘bill’. So, if you want to send to Alice 10 BEAM, your single ‘bill’ will be split into one ‘bill’ of 10 BEAM to send and another ‘bill’ of 90 BEAM to remain in your wallet, right? Well, almost: we also have to consider the transaction fee. Let’s say the transaction fee is 10M GROTH (0.1 BEAM), so in this case the ‘bill’ will be split into 3 ‘bills’ (10 BEAM to send, 89.9 BEAM to remain in your wallet and 0.1 BEAM to pay fee).

Why UTXO can be locked

**Important:** Beam Wallet app automatically selects which UTXO will be used for the transaction by trying to minimize the change you should receive as a result. This is important to understand since until the transaction is complete, the UTXOs used in the transaction cannot be used for any other transaction and do not appear in the list of ‘Available’ funds.
Example: you have 100 BEAM in two ‘bills’ (UTXO): 90 BEAM and 10 BEAM. You want to send Alice 9 BEAM. The wallet will automatically select the 10 Beam UTXO and create a transaction with 9 BEAM sent, 0.9 BEAM to remain in your wallet and 0.1 BEAM to pay fee.

This 10 BEAM UTXO will be locked until the 9 BEAM transaction completes. If Alice is currently offline, it might take a time during which you will not be able to send BEAM to anyone else. You can, of course, cancel the transaction and resend when Alice comes online.

You can split UTXO by yourself

One thing you can do is to split UTXO by sending a transaction to yourself (using you own active address). You may want to do this in the case your UTXO is too large, and you don’t want it all locked during a transaction.

**Attention:** You will pay a fee for this transaction.

Settings screen

**Note:** At the top right corner of the screen the version is displayed. It is always important to specify the version when asking for support or reporting issues.
1.4. Reporting Issues and Getting Support

To report a problem:
1. Click "Save wallet logs" and choose a destination folder for log archive.
2. Send email to testnet@beam-mw.com or open a ticket in GitHub.
3. Don't forget to attach logs archive.

Wallet folder location:
/Users/Library/Application Support/Beam Wallet
For integrated and external nodes settings see *Choosing the node connectivity mode*. When running integrated node you should specify the port on which the node will be listening on and the list of node peers.

The ‘General settings’ section allows you to change your wallet password and lock screen time. The Wallet will automatically lock to protect the funds from accidental unauthorized access to an active wallet in the set time you choose.

The ‘Report problem’ section allows you to create an archive of wallet logs and explains how to report an issue. It also shows the current location of the wallet files. See more details about reporting issues and getting support in the *Reporting Issues and Getting Support* and *Desktop Wallet Troubleshooting* sections.

**Proof of transaction**

Starting from version 2.0, receiver wallet automatically signs proof of received transaction and sends it to the sender. Proof of transaction will be added to the outcoming transaction that was successfully completed (transaction’s status is ‘sent’). Proof of transaction is not available for transaction with different status (‘expired’, ‘failed’, ‘completed’, ‘received’). Upon request, sender can observe proof of transaction following the procedure below:

1. Click outcoming transaction from the list of transactions:

   ![Wallet Interface](image)

   2. Click ‘Details’ button under Payment proof:

   3. Send payment proof code to the receiver. Copy it by clicking ‘Copy code’ button at ‘Payment proof’ form, or ‘Copy’ button in transactions details:
1.4. Reporting Issues and Getting Support
Chapter 1. Current release
Payment proof code:

4. Receiver can verify that proof is correct by clicking ‘Payment proof verification’ icon at the top right of the transactions’ list:

5. And pasting payment proof code into the form:

**Rescan wallet**

During regular operation the wallet constantly monitors the blockchain and updates the information in the wallet. However, if you suspect that your balance, transaction or UTXO status is not up to date or invalid, you can always ‘rescan’ the blockchain and update the information in your wallet with the latest state.

To perform rescan please perform the following steps:
Wallet

Payment proof verification

Paste your payment proof here

Available
14.99999999

Transactions

Date | time | Amount | Kernel ID
3/18/2019 4:19 PM | 10 BEAM | 11de66c44dee2e1c26c48831b4f78c78790ebc15c98cb69180c20ad8647f80

Details

Sender: 3e20340a4f9ef05f5e20f726524ab3d0f5bc48ef0d974e65f8b8c1669ab2e8e6025603d06f221ea4686725647371bf5321bf3d3be857f1f379900cbe92734346fa75a9400bea8e3bf1f656c44de2a457b4477c77f63ed5e55c69f19e0c20ae9547f60f4298dc3ce5e50ee0096992061621bee26058dcd87e8ff519eae51838e24df301e6c19623e44a7f5e0e23e37915e68d0eb790293d4752c6eafa860095b8c7

Receiver: 3e20340a4f9ef05f5e20f726524ab3d0f5bc48ef0d974e65f8b8c1669ab2e8e6025603d06f221ea4686725647371bf5321bf3d3be857f1f379900cbe92734346fa75a9400bea8e3bf1f656c44de2a457b4477c77f63ed5e55c69f19e0c20ae9547f60f4298dc3ce5e50ee0096992061621bee26058dcd87e8ff519eae51838e24df301e6c19623e44a7f5e0e23e37915e68d0eb790293d4752c6eafa860095b8c7

Amount: 10 BEAM

Kernel ID: 11de66c44dee2e1c26c48831b4f78c78790ebc15c98cb69180c20ad8647f80

copy details
1. In the Settings page, switch to the ‘Local node’ and click ‘Apply settings’

![Settings page](image)

2. Click on the ‘Rescan’ button and then ‘Yes’

3. Verify that your balance, transaction and UTXO state are up to date.

**Note:** If you have just started the node it may take some time for it to download updates from the blockchain and synchronize. This may result in your balance and UTXOs to be gradually updated and the node synchronizes. Please for the sync to complete before drawing any conclusions about the state of your wallet.

**Warning:** The following document is still under construction and is subject to changes.

### 1.4.2 Mobile Wallet User Guide

Beam Mobile Wallet is the simplest way to start using Beam. It is currently available for Android, the iOS app is planned to be developed later on.

**Attention:** Beam blockchain does not store transaction history and SBBS addresses. These are only stored in local database inside the wallet data folder.

Please follow the guidelines below to avoid problems with sending are receiving Beam transactions.
Rescan will update transaction and UTXO data in your wallet and get latest information from the blockchain. Are you sure?

Yes

Remote node
Ip port
us-node01.mainnet.beam.w3100

General settings
Lock screen in: never
change wallet password

Local node
Run local node
Mining threads (CPU)
0

Wallet folder location:
[Users/username/Library/Application Support/Beam Wallet]

undo changes
apply changes
1. Do not run two different wallets with the same seed at the same time
2. SBBS messages sent between wallets expire after 12 hours. You have to connect within 12 hours of the transaction initiation to receive or send the funds.
3. SBBS Addresses by default expire after 24 hours. Always use ‘never’ expiring addresses with pools and exchanges to make sure you receive payments.

**Downloading and Installing**

Proceed to install Beam Wallet Mobile App by going to the [Google Play store](https://play.google.com/store). Download the mobile wallet. Once it’s done installing, open the app and agree to the Terms of use.

**Creating new wallet**

Once you launch the wallet for the first time you will be suggested to create a new wallet.

**Generating seed phrase**

As a part of creating a new wallet, a new seed phrase will be generated for you.

---

**Attention:** Seed phrase is the most important secret you have to keep. Knowing the seed phrase enables you (or anyone else) to access all your funds.

**Attention:** Seed phrase in the Beam wallet is not linked to email, phone number or any other identifier. You will need this phrase to restore your wallet when you lose or reformat your device, or want to access your funds from another device (your mobile phone or another desktop / laptop).

**Attention:** The seed phrase is for your eyes only! Make sure no one is looking over your shoulder. For the best security always generate it on a clean air-gapped machine.

Did you wrote down your seed phrase correctly? Triple-check your handwriting again. The difference between _F_unnel and _T_unnel can be crucial when trying to restore a wallet with valuable funds in the far or near future.

Did you verified your handwriting? Now go find a safe space for the paper!

---

**Important:** Storing the seed phrase on your computer makes your funds prone to cyber attacks (read: much less secure). ‘Creative’ approaches like saving a screenshot of the wallet or your handwriting on your computer or in the cloud may sound like a good idea, but it is absolutely not recommended. If hackers get the access to your computer, network drive etc., they can can potentially steal your seed phrase by using OCR programs (which can scan pictures and transform them into plain text) and, therefore, get access to your funds.
Terms of use

PLEASE NOTE: Beam is free open-source software. While we do our best to build the most secure software we can, but we do not hold any responsibility if anything goes wrong.

Your use of Beam Wallet and Beam Blockchain is at your own risk. You may loose your money, your software may become unusable, your computer may crash and other bad things may happen.

The software is licensed under Apache 2.0 license and your use in the software is subject to the terms mentioned below.

Please read this terms of use agreement (the “Terms”) carefully as they form a binding legal agreement between you and Beam Development Limited (“Beam”, “us”, “we”, “our”), that governs the use of the digital wallet developed by Beam that allows to store, track, transfer and manage virtual currency (the “Software” or “Beam Wallet”). The term “you” refers to the person accessing or using the Software, or the entity or organisation on whose behalf such person access our Software.

1. Risks Related to the Use of the Software. Beam is not be responsible for any losses, damages or claims arising from events falling within the scope of the following: (i) mistakes made by you of any Software, forgotten passwords, payments sent to wrong Wallet Address (as defined below), and accidental deletion of Beam Wallets; (ii) software problems, e.g., corrupted Beam Wallet file, incorrectly constructed transactions, unsafe cryptographic libraries, malware affecting the Software; (iii) technical failures in the hardware of any Software, e.g., data loss due to a faulty or damaged storage device; (iv) security problems experienced by you of the Software, e.g.,
Create new wallet by generating seed phrase. If you ever lose your device, you will need this phrase to recover your wallet! Never type your seed phrase in keychains or password managers. Never save it in your local or remote folders in any form.

- Do not let anyone see your seed phrase
- Never type your seed phrase into password managers or elsewhere
- Make at least 2 copies of the phrase in case of emergency

generate seed phrase
Seed phrase

Your seed phrase is the access key to all the funds in your wallet. Print or write down the phrase to keep it in a safe or in a locked vault. Without the phrase you will not be able to recover your money.

1. garden
2. water
3. rifle
4. century
5. mutual
6. foster
7. wear
8. fantasy
9. deer
10. attend
11. approve
12. maple

→ next
Seed phrase

Your seed phrase is the access key to all the funds in your wallet. Print or write down the phrase to keep it in a safe or in a locked vault. Without the phrase you will not be able to recover your money.

Save seed phrase

Please write the seed phrase down. Do not screenshot it and save it in your photo gallery. It makes the phrase prone to cyber attacks and, therefore, less secure.
**Attention:** Always store your seed phrase in a safe and secure location (and better more than one in different geo locations). Write it on a piece of paper. Do not store electronically neither as plain text nor in any other form!

In order to ensure that you have really written down your seed phrase, you will be asked to fill in the specific words from your seed phrase in random order.

Only when you typed all the selected words correctly, you will be allowed to proceed to the next step.

**Setting wallet password**

To access your Beam Wallet, you will need to create a password. This password is not the same as the seed phrase. Seed phrase identifies a wallet and enables access to all the funds stored in it from any device. Your wallet password provides with a second security layer in case someone gains access to your computer or has stolen your wallet database file. It is important to choose a strong password.

The wallet will provide some indication of password strength for your convenience. Do not count on it, however. Choose a password that is at least 8 characters long with a combination of letters, numbers, and symbols.

**Choosing the node connectivity mode**

Beam Mobile Wallet can connect to the network through:

- Random remote node
- Specific remote node

**Note:** When running behind a firewall you can change the default port the node will be listening on, and in case of CPU mining, set the amount of mining threads. You will be probably provided at least one default peer to connect to but you can always add more peers on the Settings screen. The recommended peers are published in the list of bootstrap nodes in the downloads page on Beam official website.

Random mode allows you to automatically connect to random bootstrap node. The wallet wil have to trust the remote node for blockchain verification.

Specific remote node will automatically verify the blockchain for your wallet. If you are running your own node somewhere (or maybe even more than one) and want to connect specifically to it, use the option by providing the IP and port the node is listening on.

**Synchronizing the wallet**

Once the wallet is connected, it synchronizes with the current blockchain state from the network. Be patient when running with the integrated node: the data downloading process might take some time. The wallet will first download and validate the latest Macroblock and then all the rest of the blockchain.

**Main screen**

Once your wallet is created, the main screen will show up.

On the top left, the hamburger button (the three lines) provides navigation between different wallet screens, currently: Wallet (or Main Screen), Address book, UTXO, Dashboard, Notification, Help and Settings.
Confirm seed phrase

To ensure the seed phrase is written down correctly, please fill in the specific words below:

3 garden  7 wear
2 water  12 maple
4 century  6 foster

→ next
Create password to access your wallet

Password

Strong password needs to meet the following requirements:
• the length must be at least 10 characters
• must contain at least one lowercase letter
• must contain at least one uppercase letter
• must contain at least one number

Confirm password

→ start using your wallet
Creating wallet
1.4. Reporting Issues and Getting Support

Transaction history is empty.
Chapter 1. Current release

Wallet
Address book
UTXO
Settings
Wallet status indicator

The main screen of the wallet shows the current balance as well as the transaction history and statuses. On the left, under the Beam logo, there is a toolbar that provides navigation between different wallet screens such as Main Screen, Addresses Screen, UTXO Screen and Settings Screen.

In the top left corner of the Main Screen, under ‘Wallet’, you see connection status indicator.

Green indicator means that the node is up and running connected to peers.

Red means it is either unable to connect to the node or the node has a problem you can help to solve.

In brackets, the indicator also specifies the network to which the wallet is connected:

- **Mainnet**: the network with real money and actual transactions.
- **Testnet<number>** (such as Testnet3, Testnet4 etc): staging environments for trying new features in our wallet, node and miner software. For advanced users only.
- **Masternet**: new features under development, if you see this name in your wallet it means that you are very early adopter or Beam code contributor otherwise you’ve probably arrived to the wrong place.

**Attention:** Since Mainnet is the default network for the vast majority of Beam users, nothing is written by the online status.

Finally, node connectivity node is displayed (ie. random remote node or specific node).

Financial transactions

‘Send’ and ‘Receive’ buttons at the top right corner help us with *Sending BEAM* and *Receiving BEAM*.

Main screen password

In the future, the screen will pop-up automatically after you open the application and type in your password.

Receiving BEAM

Before starting to receive BEAM for the first time, please read first about what *Address* is.

Here is how the process of receiving BEAM looks like from a Receiver’s perspective:

- Generate an address
- Send your address to the Sender person *over a secure communication channel*
- Both Sender and Receiver’s Wallet must be online at the same time to complete a transaction.

It’s possible to reuse an address that already exists, more on that later.

Generate an address

Proceed to the main screen and click the blue ‘Receive’ button at the top right corner.

This will open the receive screen.

Copy and paste the newly generated Beam address to send to Sender over a *secure communication channel*. 
Enter your password to access the wallet

Password

open your wallet

login to another wallet

Forgot password?
Wallet

online

receive

send

Available

0

Transactions

Transaction history is empty.
Receive

online

My address

Expires

in 24 hours

Comment

Comments are local and won't be shared.

Your transaction token:

176dan89jksasdg21skaw9q7g176dan
89jksasdg21skaw9q7g176dan89jksas
dg21skaw9q7g176da89jksasdg21ska
w9q7g176dan89jksasdg21skaw9q7g

Send this token to the sender over secure channel
A new Beam address is generated every time the ‘Receive’ screen is opened. By default, the address is valid for 24 hours. It is best to give the Sender your address closest to the time they will be sending BEAM so the address does not expire.

You can explicitly set the expiration time to ‘Never’ for this address only by selecting the value in the ‘Expires’ drop down.

**Warning:** The permanent addresses that never expire are only useful in limited special cases, like working with mining pools or exchanges. For ultimate privacy, do not use this option for regular transactions, rather always create a new address for each transaction.

**Attention:** You have to click ‘Copy address button in the screen for the address to become active.

**QR code**

If the Sender uses a mobile app, he can quickly scan the QR code from your mobile instead of receiving, copying and pasting the alphanumeric address.

**Comment**

You can add a comment when creating the receiving address. The comment is never sent to the network, it is only visible inside your wallet and is used for internal bookkeeping only.

The comment can be seen on the *Address screen* and in the extended transaction view.

**Sending the address**

**Attention:** When sending the address make sure you use a secure communication channel.

**Attention:** Make sure the entire address is sent to the Sender as it’s longer than it appears on the screen. *Don’t forget* to double check the value in whichever messenger app of your choice because viruses and malware on your computer may change your address while it’s in the clipboard.

**Completing the transaction**

Once Sender initiates the transaction, you will see on the new transaction appear in the transaction list on the main screen. The amount sent will also appear on the ‘In progress’ box.

Normally, a transaction will pass through the following stages:

- **In Progress** - the phase when the transaction is being created by the Sending and Receiving wallets.
- **Confirming** - the phase after the transaction was sent to the nodes, but before it is mined.
- **Completed** - the phase after the transaction is fully mined and confirmed.

Once transaction is complete, the available balance will be updated and the ‘In Progress’ box will disappear.
Your transaction token:

176dan89jksasdg21skaw9q7g176dan
89jksasdg21skaw9q7g176dan89jksas
dg21skaw9q7g176da89jksasdg21skaw9q7g

Send this token to the sender over secure channel
Your transaction token:

176dan89jksasdg21skaw9q7g176dan89jksasdg21skaw9q7g176dan89jksasd
g21skaw9q7g176da89jksasdg21skaw9q7g176dan89jksasdg21skaw9q7g

Scan this QR code or send this token to the sender over a secure channel
1.4. Reporting Issues and Getting Support

Your transaction token:

176dan89jksasdg21skaw9q7g176dan
89jksasdg21skaw9q7g176dan89jksas
dg21skaw9q7g176da89jksasdg21ska
w9q7g176dan89jksasdg21skaw9q7g

Send this token to the sender over secure channel
Wallet

online

receive  send

Available

54 B

Transactions

Receive BEAM

1 Jan 2019  |  3:46 PM

completed
Sending BEAM

Before starting to send BEAM, please read first about what :ref:`Address` is.

Here is how the process of sending BEAM looks like from a Sender’s perspective:

- Receive the address the funds should be sent to
- Send BEAM to Receiver
- Stay online until Receiver confirms the transaction

Receiving the address

**Attention:** Make sure that the address is received untampered by using a secure communication channel.

**Attention:** When copying the address to the Beam Wallet app please verify visually that the address in the wallet looks exactly like the address in the secure messaging app, because viruses and malware on your computer may change your address while it’s in the clipboard.

Sending funds

In order to send BEAM, you will need to click the magenta ‘Send’ button at the top right corner. This will open the Send screen.

Make sure you have the correct address and paste the Receiver’s Beam address in the ‘Send To’ field.

To help to identify the transaction, you may also choose to fill in the optional Comment field. The comment will remind you what or who the transaction is for. The comment is stored locally, thus it will only be visible in your wallet for bookkeeping purposes.

The comment can be seen on the Address screen:

The comment is also displayed in the extended transaction view on Main Screen:

Select the transaction amount in BEAM you want to send. Transaction amount is in BEAM and may contain fractional values such as 1.25 BEAM or 11.3 BEAM and the like. Keep in mind you also have to pay a transaction fee, hence the amount to send plus the fee must be equal to or less than the available balance.

Transaction fees are specified in GROTH (100 millionths of BEAM). Amount of fees you need to pay depends on the current status of the network and average fee sizes. Simply said, the higher transaction fee will help miners to prioritize your transaction. To determine the current average fee size use Beam Blockchain Explorer.

You can see the remaining amount of BEAM in your wallet and the change that will be received after the transaction.

After you click ‘Send’ you will see a confirmation with the most important transaction details:

Completing the transaction

Once you confirm, the transaction is sent to the Receiver’s wallet. If Receiver’s wallet is currently offline or if the network is loaded, you might see the transaction appear ‘In Progress’ on your transaction list. Once the transaction is complete, it will be sent to the nodes and shown as ‘Confirming’.
1.4. Reporting Issues and Getting Support
 Beam Documentation, Release 2019

Send

online

Send to
167jbfsdjfkfkk39902mnsdnkbkjadavfd39nas7877qwb

Send amount
30

Transaction fee
12 hours | 10 GROTH
48 h | 1 GROTH 20 m | 20 GROTH

Comment

Jane

Comments are local and won't be shared.

Change:
(Will be locked till the transaction will succeed or expire)
Sending BEAM
1 Jan 2019 | 6:13 PM
-30 confirming

General transaction info

Contact: 167jbfsdjklk39902mnnsdnkbbjadavfd39nas7877qwb
My address: 1Cs4wu6pu5qCZ35bSLNVzGyEx5N6uzbg9t
Transaction status: confirming
Currency amount: For the day of transaction
Transaction fee: 10 GROTH
Comment: Jane
Kernel ID: 1Cs4wu6pu5qCZ35bSLNVzGyEx5N6uzbg9t1Cs4wu6pu5qCZ35bSLNVzGyEx5N6uzbg9t
Send

online

Send to
167jbfsdjflk39902mnsdnkbbkjadafd39nas7877qwb

Send amount
30

Transaction fee
12 h | 1 GROTH

48 h | 0 GROTH | 20 m | 20 GROTH

Comment
Jane

Comments are local and won't be shared.

Change:
(will be locked till the transaction will succeed or expire)
Send to
167jbfstdjflk39902mnsdnkbkjadavfd39nas7877qwb

Recipent: 167jbfstdjflk39902mnsdnkbkjadavfd39nas7877qwb

Amount: 30 BEAM

Transaction fee: 10 GROTH

Comments are local and won't be shared.
Wallet

Available

19.9999990 B

In progress

-30 B

Transactions

Sending BEAM

1 Jan 2019 | 6:13 PM

-30 B

confirming
**Note:** While a transaction is in ‘In Progress’ you can cancel it by clicking on the dropdown to the right of the transaction row and then select ‘Cancel’. The other party will receive notification that the transaction was either ‘Canceled’ or ‘Failed,’ and funds plus fee that were allocated for this transaction will become available again. It is not possible to cancel a transaction in ‘Confirming’ or ‘Completed’ states.

**Warning:** If your transaction appears as ‘In Progress’ for a long time, it means the Receiver is not online.

**Attention:** If the transaction was not sent to the nodes, for any reason, it will expire after 1440 blocks, or roughly 24 hours. This is done to avoid a situation in which UTXO is locked forever.

**Restoring funds**

This process allows you to restore your funds directly from the blockchain. It is useful in the scenarios such as:

- You’ve got a new device and would like to use your wallet on it
- You forgot your local password and can’t access your funds

**Attention:** With Beam, only funds are stored on the blockchain. Everything else, such as your active addresses, contacts or transaction history can’t be restored.

**Attention:** In the current mobile wallet implementation restoring funds is only implemented in the Beam Desktop Wallet App. Here is what you need to do:

- Proceed to your desktop wallet app
- Restore your funds on desktop using the mobile wallet seed phrase
- Create the new wallet on mobile
- Transfer the restored funds to your newly created mobile wallet

**Address**

Let’s define the meaning of address in the BEAM ecosystem: BEAM is always sent from one address to another. Both sending and receiving addresses are alphanumeric tokens that uniquely define the transaction endpoints.

A person can create as many addresses as required. The address creation process is explained in *Receiving BEAM*.

*Example:* There are two wallets: one belongs to you and the other belongs to Alice. You have created one address to receive money from Alice and another address to send money to Alice. Since you can generate multiple addresses, Alice will never know that she’s receiving money from the same person that she is sending the money to (unless you want her to know).

*Example:* There are three wallets: one belongs to you, one belongs to Alice and another belongs to Bob. You have created one address to receive money from Alice and another address to receive money from Bob. Alice and Bob will never know that they are sending money to the same person.

Same address can be used for sending and receiving money.
Example: you have created an address to send money to Alice. Alice can see the address the money came from and can send money to the address back to you.

**Attention:** For ultimate privacy, it is advised to have a dedicated address for every transaction (ie. for both Sending or Receiving).

**Attention:** Although not recommended, an address can be reused until it had reached its expiration (24 hours since when it was generated).

Example: Imagine you’ve created the address with expiration interval of 24 hours and immediately sent it to Alice. In the next 24 hours, Alice will be able to send BEAM to you as many times as she likes, reusing the same address of yours.

**Warning:** Reusing same addresses, created with longer expiration interval, can be convenient yet the tradeoff of decreased privacy should be kept in mind.

### Address screen

The screen lists all the addresses that were used in all incoming and outgoing transactions. All the data in this screen is only stored locally in your wallet and is not related to the blockchain in any way.

**Note:** For your own privacy, a new address is generated for each outgoing transaction. Yes, you heard that right! That address can be seen in your active or expired addresses list.

Upon Beam Wallet mobile app installation, a single address is created by default. The address has a default expiration time of 24 hours.

You can always create a new one by using Receive Screen. You can see all your active addresses in the ‘My Active Addresses’ tab.

Once an expired address becomes inactive, it cannot be used in any incoming or outgoing transaction. Each address has a default expiration time of 24 hours, including the default address. An expired address becomes inactive and you won’t be able to be use it again. You can see the expired addresses listed under ‘My Expired Addresses.’

When you’ll click the three dots located to the right of any address, the menu with additional address actions will open. You can manually edit or delete any address choosing ‘Edit’ or ‘Delete’ from the menu.

You can manually expire any active address by choosing ‘Expire address’ from the drop down menu.

For privacy concerns, you can manually delete any address by clicking the three dots located to the right of the address and choosing ‘Delete address’ from the drop down menu.

In the ‘Contacts’ tab, you can see every address that sent you BEAM or to which you’ve sent BEAM to.

### UTXO

UTXO (Unspent Transaction (TX) Output) is like a banknote of a specific amount. Simply said, if BEAM is the currency, any UTXO can be considered a ‘bill’. You can have multiple ‘bills’ in your wallet at the same time.
Addresses

online

MY ACTIVE ADDRESSES

Default

167jbf...877qwb
Expires: 2 Jan 2019  |  3:46 PM

MY EXPIRED ADDRESSES
Addresses

online

Addresses

MY EXPIRED ADDRESSES

Default

167jbf...877qwb

Expired: 2 Jan 2019 | 3:46 PM
Edit address

Address ID: 1Cs4wu6pu5qCZ35bSLNVzGyEx5N6uzbg9t
Expires: 3 Jan 2019 | 3:45 PM (GMT +3)

Comment
Business vendors
Addresses

online

MY EXPIRED ADDRESSES

CONTACTS

Default

167jbf...877qwb
**UTXO screen**

On the technical level, in Beam, like in most other cryptocurrencies, your balance emerges as a result of multiple incoming and outgoing transactions. Each transaction uses some existing inputs and creates new outputs. All the outputs controlled by the wallet are shown in the UTXO screen.

The type of UTXO can be:

- **Regular** - UTXO received as a result of a transaction. It is immediately available for spending
- **Change** - UTXO received as a result of change from a transaction. It is immediately available for spending
- **Transaction fee** - Fees received as a result of mining a block which contain transactions
- **Coinbase** - UTXO you have mined. It has maturity of 3 hours (240 blocks) and will not be immediately seen in Available tab

**UTXO details**

Click on the specific UTXO you want to see more details of.

**UTXO in sending BEAM screen (explained by example)**

Assume that you have 100 BEAM in a single ‘bill’. So, if you want to send to Alice 10 BEAM, your single ‘bill’ will be split into one ‘bill’ of 10 BEAM to send and another ‘bill’ of 90 BEAM to remain in your wallet, right? Well, almost: we also have to consider the transaction fee. Let’s say the transaction fee is 10M GROTH (0.1 BEAM), so in this case the ‘bill’ will be split into 3 ‘bills’ (10 BEAM to send, 89.9 BEAM to remain in your wallet and 0.1 BEAM to pay fee).

**Why UTXO can be locked**

*Important:* Beam Wallet app automatically selects which UTXO will be used for the transaction by trying to minimize the change you should receive as a result. This is important to understand since until the transaction is complete, the UTXOs used in the transaction cannot be used for any other transaction and do not appear in the list of ‘Available’ funds.

Example: you have 100 BEAM in two ‘bills’ (UTXO): 90 BEAM and 10 BEAM. You want to send Alice 9 BEAM. The wallet will automatically select the 10 Beam UTXO and create a transaction with 9 BEAM sent, 0.9 BEAM to remain in your wallet and 0.1 BEAM to pay fee.

This 10 BEAM UTXO will be locked until the 9 BEAM transaction completes. If Alice is currently offline, it might take a time during which you will not be able to send BEAM to anyone else. You can, of course, cancel the transaction and resend when Alice comes online.

**You can split UTXO by yourself**

One thing you can do is to split UTXO by sending a transaction to yourself (using you own active address). You may want to do this in the case your UTXO is too large, and you don’t want it all locked during a transaction.

**Attention:** You will pay a fee for this transaction.
UTXO details

23.99999

Transaction ID: 1Cs4wu6pu5qCZ35bSLNVzG
  yEx5N6uzbg9t

UTXO type: coinbase

Completion time: 5 hours

Contact: magic_stardust16
  1Cs4wu6pu5qCZ35bSLNVzG
  yEx5N6uzbg9t

UTXO history

<table>
<thead>
<tr>
<th>Date</th>
<th>time</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Jan 2019</td>
<td>3:45 PM</td>
<td>1h67sg...87yhd6</td>
</tr>
</tbody>
</table>
UTXO

online

ACTIVE

23.99999

available

8jdu7we232ihsjwbdy62e2n3jr3o9bda

3.99999

available

8wnsuZpxdsgwqiwnyqsgLSDD723T

9

spent

9f8sn49jllsnw926sn6s8s72hs&

40

available

98hskasybs6g28sdjsbn9shiqod

5

spent

958sn34fjllw086an6dsn72hs&
Send

online

Send to
uy83hg6pu5qCZ35bSLNVzGyEx5Njd8ush

Send amount

10

Transaction fee

12 hours | 10000000 GROTH

48h
100000 GROTH
20m
100000000 GROTH

Comment

Comments are local and won't be shared.

Change:
(will be locked till the transaction
1.4.3 Command Line Wallet User Guide

The purpose of this document is to describe the process of setting up Beam node and command line wallet.

**Attention:** Beam blockchain does not store transaction history and SBBS addresses. These are only stored in local database inside the wallet data folder.

Please follow the guidelines below to avoid problems with sending are receiving Beam transactions.

1. DO NOT run several wallet processes on the same wallet.db file.
2. Do not do listen and send at the same time using CLI wallet
3. Do not copy the wallet.db to another machine and run another wallet simultaneously using the same wallet database
4. Do not run two different wallets with the same seed at the same time
5. SBBS messages sent between wallets expire after 12 hours. You have to connect within 12 hours of the transaction initiation to receive or send the funds.
6. SBBS Addresses by default expire after 24 hours. Always use ‘never’ expiring addresses with pools and exchanges to make sure you receive payments.

**Getting Started**

Beam software runs on all operating systems: Linux, Mac OS and Windows. Before you start please verify that your operating system is supported by reviewing the [Supported Platforms](#).

All examples in this section are formatted for Linux / Mac platforms. If you are using Windows please substitute ./beam-wallet with beam-wallet.exe and run your commands using Windows Command Prompt.

**Creating new wallet**

In order to create a new wallet run:

```bash
./beam-wallet init
```

You will be prompted to enter the Wallet Password, which is used to protect the wallet database

**Warning:** Choose strong password for the wallet and keep it secret

Anyone who knows your wallet password and can access your machine the wallet is stored on, will be able to spend **all your funds** and will have an access to all the metadata stored in the wallet, including transaction history!

Sample output for the `init` operation will look something like this:

```
$ ./beam-wallet init
I 2018-12-23.15:24:29.461 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.15:24:29.462 starting a wallet...
Enter password: ****************
I 2018-12-23.15:24:32.524 Generating seed phrase...
Generated seed phrase:
```

(continues on next page)
IMPORTANT

Your seed phrase is the access key to all the cryptocurrencies in your wallet. Print or write down the phrase to keep it in a safe or in a locked vault. Without the phrase you will not be able to recover your money.

I 2018-12-23.15:24:32.728 wallet successfully created...
I 2018-12-23.15:24:32.750 New address generated:
14a38140d8e66be9b8f1e8d770161fd33e35f7000053147b5a0f6a8378926b956
I 2018-12-23.15:24:32.750 label = default

The Rules signature is a hash of current node configuration which is used to determine compatibility between different versions of nodes and wallets.

Generated seed phrase is the **Seed Phrase**.

**Warning:** Copy the seed phrase to a secure location and keep it safe.

Seed phrase is the **most important secret** you need to keep to protect your funds. Anyone knowing the seed phrase will be able to control all your funds regardless of any other information. When generating new wallet each and every time, the safest scenario would be to make it on a secure air-gapped machine in a private environment and always keep the seed phrase in a secret and protected place.

The following line indicates that a new temporary **SBBS** address has been generated. This address is valid for the next 24 hours and can be used to receive coins. To generate new addresses see ‘Creating new receive address’ section below.

After wallet initialization is succeeded a **wallet.db** file is created in the same folder the wallet was run at. **wallet.db** is the wallet database file which is encrypted with the Wallet Password and contains the entire transaction history, keys and all the rest of the wallet metadata. If this file is deleted or lost, for any reason, you can always restore your funds using the Seed Phrase, however you will lose all transaction history and any additional metadata stored in the wallet database. To understand how to backup and restore the **wallet.db** file please check the **Backup and Restore** page.

In addition to the **wallet.db** file, you will see the **logs** folder. A new log file is created every time you run the CLI wallet. Please attach logs to any support request you might send. See Reporting Issues and Getting Support section, for more details.

**Restore a wallet from a Seed Phrase**

For all the restore procedures see Restore CLI wallet from Seed Phrase

**Exporting miner key**

To generate a secret key used by the miner to attribute mining rewards to your wallet run the following command:
./beam-wallet export_miner_key --subkey=<integer miner id, i.e 1,2,3...>

You will be prompted for the wallet password

The sample output for this command should look like this:

```
$ beam-wallet.exe export_miner_key --subkey=1
I 2018-12-23.16:36:04.306 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.16:36:04.307 starting a wallet...
Enter password: *******************
Secret Subkey 1: OVBSdWQlOV3uC6bLXRDJqyDfxWSuzdA4jEGRAZ1zhy4gA3/
→KcBTEdcmN5vN0v0QrBwW1tId1xqyPFzFDFdaVYzPUDoXjggUE= 
```

It is important to keep the Miner Key secret since anyone who knows the miner key will be able to spend all the rewards mined by that miner.

**Exporting owner key**

The purpose of the Owner Key is to allow all nodes mining for you to be aware of all mining rewards mined by other nodes so that you would only need to connect to one node to collect all the rewards into your wallet. While in most other cryptocurrencies this is done by simply mining to a single address you control, in Mimblewimble it is not as simple since there are no addresses and the mining rewards should be coded with unique blinding factors which are deterministically derived from the Master Key, and then tagged by the single Owner Key.

Owner Key should be kept secret. Owner Key does not allow to spend coins, however it will allow to see all coins mined for you by all miners that use this Owner Key.

To export the Owner Key run the following command:

```
./beam-wallet export_owner_key
```

You will be prompted for the wallet password

Sample output for this command should look like this:

```
$ ./beam-wallet export_owner_key
I 2018-12-23.16:53:04.973 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.16:53:04.974 starting a wallet...
Enter password: *
Owner Viewer key: dmVxtRCM3BH1VakviSB/
→XY86DsCwKuWDLKk51eLD1ibqMeL2fZ3l72dQx3E6oXbKt1dqZz/1c5stTCSz9M1bDJdYUF4DG/ZaIuHHszI/
→H9wDmNDVboUdNtC/1Z/haWr9JxeIDtRSDBN+xpUbv
```

**Printing the wallet info**

To print the current status of your wallet, run the following command:

```
./beam-wallet info
```

You will be prompted for the wallet password

A sample output for this command should look like this:

```
I 2018-12-23.17:56:19.368 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.17:56:19.369 starting a wallet...
Enter password: *
```
I 2018-12-23.17:56:21.144 wallet sucessfully opened...

---Wallet summary---

Current height............8353
Current state ID..........72329a2efa2ddad4

Available..................300 beans
Maturing..................0 groth
In progress...............0 groth
Unavailable...............0 groth
Available coinbase.......0 groth
Total coinbase............0 groth
Available fee.............0 groth
Total fee..................0 groth
Total unspent.............300 beans

<table>
<thead>
<tr>
<th>id</th>
<th>Beam</th>
<th>Groth</th>
<th>height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1545571472000001</td>
<td>300</td>
<td>0</td>
<td>8347</td>
</tr>
<tr>
<td>8351</td>
<td></td>
<td>norm</td>
<td></td>
</tr>
</tbody>
</table>

It is also possible to see the transaction history using the --tx_history flag

./beam-wallet info --tx_history

You could also see the details of transaction using this command:

./beam-wallet tx_details --tx_id=<txid>

---Receiving BEAMs---

To receive BEAMs you need to connect to a specific node by running the following command:

./beam-wallet listen -n <node address and port, ex: 127.0.0.1:10000>

You will be prompted for the wallet password

A sample output for this command should look like:

I 2018-12-23.17:07:55.526 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.17:07:55.527 starting a wallet...
Enter password: ***************
I 2018-12-23.17:07:58.076 wallet sucessfully opened...
I 2018-12-23.17:07:58.078 WalletID...
...14a38140d8e66be9b8f1e8d770161fd33e35f7000053147b5a0f6a83178926b956 subscribes to...
...BBS channel 20
I 2018-12-23.17:07:59.297 Sync up to 8304-2dc4e5a393d6774b
I 2018-12-23.17:07:59.318 Current state is 8304-2dc4e5a393d6774b

Once launched, the wallet will listen to updates from the server and any incoming transactions on the advertise SBBS address.

To receive funds you should send the address to the sending party via any available secure channel (Email, Telegram etc.)

When funds are sent you will see the incoming transaction in the wallet logs and on the screen. It should look similar to:

1.4. Reporting Issues and Getting Support
Sending BEAMS

To send beams you need to run the following command:

```
./beam-wallet send -n <node address and port, ex: 127.0.0.1:10000> -r <sbbs address> -a <amount (in Beams), ex: 11.3> -f <fee (in Groth), ex: 0.2>
```

Note: 1 Groth equals 10^-8 Beam

The wallet log should look similar to something like:

```
$ ./beam-wallet send -n 172.104.249.212:8101 -r
-a 14a38140d8e66be8b8f1e8d770161fd3e35f7000053147b5a0f6a83178926b956 -a 10
I 2018-12-23.18:05:49.037 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.18:05:49.038 starting a wallet...
Enter password: *
I 2018-12-23.18:05:50.725 wallet sucessfully opened...
I 2018-12-23.18:05:50.726 WalletID 14a38140d8e66be8b8f1e8d770161fd3e35f7000053147b5a0f6a83178926b956 subscribes to BBS channel 20
I 2018-12-23.18:05:50.775 [b21f08337dd94603bb038c82c1888eac] Sending 10 beams (fee: 0 groth)
I 2018-12-23.18:05:50.986 [b21f08337dd94603bb038c82c1888eac] Invitation accepted
```
2. in the `send` command, add `--utxo` parameter and specify a comma separated list of utxo ids:

```
./beam-wallet send -n <node address and port, ex: 127.0.0.1:10000> -r <sbbs address> -a <amount (in Beams), ex: 11.3> -f <fee (in Groth), ex: 0.2> --utxo=<comma-separated list of utxo ids>
```

### Cancelling and deleting the transaction

Sometimes due to unsuccessful transaction or in different cases you need to cancel transaction. It is possible following the procedure below:

1. Print the list of transactions using:

```
./beam-wallet info --tx_history
```

2. Get the id of the transaction you need and run:

```
./beam-wallet cancel_tx --tx_id=<txid, ex: f1e11512141a4f59b1c59ab1386ea84> -n <node address and port, ex: 127.0.0.1:10000>
```

Also you could delete useless transaction via first step and following command:

```
./beam-wallet delete_tx --tx_id=<txid, ex: f1e11512141a4f59b1c59ab1386ea84> -n <node address and port, ex: 127.0.0.1:10000>
```

### Creating new SBBS address

In order to create new SBBS address, run the following command:

```
./beam-wallet new_addr --expiration_time=never124h --comment="some comment"
```
You will be prompted for the wallet password

Sample output from this command should look like this:

```
I 2018-12-23.18:16:44.112 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.18:16:44.113 starting a wallet...
Enter password: *
I 2018-12-23.18:16:45.392 New address generated:
646a773da4d4651f35fd75ca958b7859e89d8d8382b8155773bd396e2cc49cca
```

Print list of all addresses

To print the entire list of addresses use the following command:

```
./beam-wallet address_list
```

A sample output for this command will look something like this

```
I 2019-02-25.19:41:26.839 Beam Wallet 1.2.4419 (mainnet)
I 2019-02-25.19:41:26.841 starting a wallet...
Enter password: *
I 2019-02-25.19:41:27.718 wallet sucessfully opened...
Addresses

<table>
<thead>
<tr>
<th>comment</th>
<th>address</th>
<th>active</th>
<th>expiration date</th>
<th>created</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14e191aebace13b14e3ab41382280baff288faa312545ead1albcfa3adaea6ff</td>
<td>false</td>
<td>2019.02.25 12:34:07</td>
<td>2019.02.24 12:34:07</td>
</tr>
<tr>
<td></td>
<td>12508d77079a41ca9929ed33b965758f261030a766d3bfcf0524celd21f55b88dc8ff</td>
<td>false</td>
<td>2019.02.20 12:32:11</td>
<td>2019.02.19 12:32:11</td>
</tr>
</tbody>
</table>
```
Change address expiration

It is possible to change address expiration period to ‘never’ using the following command:

```
./beam-wallet change_address_expiration --
    --address=14e191aebace13b14e3ab41382280baff288faa312545eadd1abcfa3adaeac6ff
```

For specific address.

To change all existing addresses in the wallet just omit the `--address` parameter

You could also choose what specific period it would be:

```
./beam-wallet change_address_expiration --address=<sbbs address> --expiration_time=(never|24h)
```

There are two options: never or 24 hours.

Export and import SBBS addresses and transaction history

Sometimes when upgrading the wallet or restoring from seed phrase you need to reimport the list of SBBS addresses and transaction history (the data) from the previous wallet. To do that use the commands below:

To export the data:

```
./beam-wallet export_data --file_location=<full path to addresses file, for example: C:\Users\user\addresses.dat>
```

To import the data

```
./beam-wallet import_data --file_location=<full path to addresses file, for example: C:\Users\user\addresses.dat>
```

**Note:** It is important that imported addresses were originally created by the wallet with the SAME seed phrase. Only addresses matching the wallet seed phrase will be imported. Other address will not be imported as shown in the screenshot below.

Proof of transaction

Starting from 1.1.4194 version, receiver wallet automatically signs proof of received transaction and sends it to the sender. Upon request, sender can generate proof of transaction following the procedure below:

1. Print the list of transactions using:

```
./beam-wallet info --tx_history
```

2. Get the id of the transaction we need and run:
./beam-wallet payment_proof_export --tx_id=<txid>

Sample output of the command above should look something like:

```
I 2019-01-14.14:40:37.465 Payment tx details:
Sender: 4bd0ca080bd8c3ec4b3061bf591aa34266f0649a7c151c6777ffe492f15e09768
Receiver: ebb27b5501213c84eb212ea276e8ced74f540fbccebf0f4c1c2da2c5108188651a1
Amount: 6 groth
KernelID: 4ac2f195ce9056c171fd0cd41e8a02dc90bb72861b2e03fbbb5942e5e63d1a
```

3. Send the contents of exported form only (proof) to the receiver, our case it will look like this:

```
I 2019-01-14.14:40:37.465 Exported form: 000000000000004bd0ca080bd8c3ecd4b3061bf5
91aa34266f0649a7c151c6777ffe492f15e097680000000000000eebb27b5501213c84eb212ea27
6ee8ced74f540fbccebf0f4c1c2da2c5108188651a1864ac2f195ce9056c171fd0cd41e8a02dc90bb
72861b2e03fbbb5942e5e63d1a7728a2954a10d3bfb9938f0c17509a6a0e870c6bb22ff2d1297f3
daef754592b00e84c6b3c9ea3e3ad9bc43661b6dcef7dbd818ccc927071d75b429697e8492653
```

4. Receiver can verify that proof is correct by running:
```bash
./beam-wallet payment_proof_verify --payment_proof=<proof>
```

**Note:** Sender can require receiver to always send proof of transaction by using `--payment_proof_required=1`. Please note that this will prevent working with older wallets.

---

**Cold Wallet**

To use the wallet in ‘cold’ mode you need to initialize it with `--cold_wallet` flag.

```bash
./beam-wallet init --cold_wallet
```

This command will create two databases: wallet.db and wallet.db.private.

---

**Sending from cold wallet**

Pre-conditions: Make sure the cold wallet is synced. In order to do so, follow the next steps:

1. Copy the wallet.db file to the “hot” wallet’s data folder.
2. Launch the “hot” wallet and wait till it’s synced.
3. Stop the “hot” wallet, copy the wallet.db file into the “cold” wallet folder.
4. Launch the “cold” wallet for listening.

```bash
./beam-wallet listen --cold_wallet
```

Now as the “cold” wallet is synced, proceed with the next steps:

1. In the cold wallet run the command:

   ```bash
   ./beam-wallet send -a <amount> -r <receiver address> -f <fee> --cold_wallet
   ```

   **Note:** Here is no need in node address in this case.

2. Copy wallet.db file to “hot” wallet’s data folder.
3. Launch “hot” wallet. It should send encrypted message to the node, also he may get encrypted message back.
4. Stop “hot” wallet, copy wallet.db file into “cold” wallet folder.
5. Launch “cold” wallet for listening `beam-wallet listen --cold_wallet` it should create a signed transaction kernel.
6. Copy wallet.db from “cold” to “hot” new transaction should go to the node and got confirmed.
7. Copy wallet.db from “hot” to “cold” “cold” wallet should have actual balance and transactions statuses.

---

**Receiving to cold wallet**

1. Generate new address in “cold” wallet and send it to the sender.
2. Copy wallet.db to “hot” wallet.
3. Launch “hot” wallet. Note there will be no new transactions, since “hot” wallet cannot decrypt incoming messages.
4. Stop “hot” wallet. Copy wallet.db from “hot” to “cold”.
5. Launch “cold” wallet for listening, it should get new transaction and accept it.
6. Copy wallet.db from “cold” to “hot”.
7. Launch “hot”, wait until new transaction becomes completed.
8. Copy wallet.db “hot” to “cold” balance and transactions statuses should be correct.

Rescan wallet

During regular operation the wallet constantly monitors the blockchain and updates the information in the wallet. However, if you suspect that your balance, transaction or UTXO status is not up to date or invalid, you can always ‘rescan’ the blockchain and update the information in your wallet with the latest state.

In order to rescan the CLI wallet please follow the steps below:

1. Run a node with your ‘owner’ key and make sure it has completed the synchronization with the network. See Exporting owner key

   ```
   ./beam-node --peer=<ip or url of the peer> --owner_key=<your owner key>
   ```

2. Run ‘rescan’ command as follows:

   ```
   ./beam-wallet rescan -n <ip:port of the node with the owner key>
   ```

3. Run ‘listen’ command to get updated information from the node

   ```
   ./beam-wallet listen -n <ip:port of the node with the owner key>
   ```

4. Wait for the wallet to synchronize and check that balance and transactions were update using ‘info’ command

   ```
   ./beam-wallet info
   ```

1.4.4 Beam Node User Guide

General

Beam Node is an essential part of the Beam blockchain and is responsible for validating transactions and blocks. It runs on all platforms: Linux, Windows and Mac (for detailed list of supported and tested platforms please see Supported Platforms)

Beam Node can be run in either Mining or Validating mode.

Mining mode

Beam Node supports External Miner via Statum Server API for mining Beam:

Beam Node provides built in support for Startum API allowing to connect multiple external mining clients to a single node. (see :ref: user_beam_stratum_server for more details). To start the node with stratum server use stratum_port and stratum_secrets_path parameters. Stratum clients can be run together with the Internal Miner
Mining keys management

In order for the mining node to be able to attribute mining rewards to your wallet, it needs a special secret mining key. The mining key is derived from the primary secret key by running `export_miner_key` command with `--subkey=<node id>` parameter in the Beam CLI Wallet (See Command Line Wallet User Guide for more details). You can generate a multiple separate mining keys for different mining nodes.

Optionally, in order to allow each mining node to be able to see all rewards mined by all your mining nodes Beam provides an additional option called owner_key. Owner key is a secret view key, it can not be used to spend coins, just to identify your mining rewards regardless of which node was used to mine it. Owner key is derived from primary secret key as well using the same `key_export` command, but without additional parameters.

Both keys are protected using Wallet Password, which should also be provided.

Validating mode

By default (without `--miner_type` flag) Beam Node is run in validating mode, meaning that mining is disabled. Validating nodes are still very important for the overall health and safety of the network since they:

1. Help in propagating transactions and blocks through the network
2. Relay SBBS messages to enable Wallet to Wallet communication.
3. Serve as Dandelion Stem relays to improve P2P level security

If possible, always prefer running a local node either with or without mining!

Node Settings

Beam Node allows to provide the settings via command line or using a configuration file called beam-node.cfg and located in the same folder as Beam Node binary.

Command line parameters override configuration file settings

The configuration file is loaded automatically and sets all parameters that were not provided via command line. To reload configuration file after a change you should manually restart Beam Node.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description &amp; Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--port</code></td>
<td>Port to start the server on</td>
</tr>
<tr>
<td></td>
<td><code>port=10000</code></td>
</tr>
<tr>
<td><code>--log_level</code></td>
<td>Log level [info</td>
</tr>
<tr>
<td></td>
<td><code>log_level=info</code></td>
</tr>
<tr>
<td><code>--file_log_level</code></td>
<td>File log level [info</td>
</tr>
<tr>
<td></td>
<td><code>file_log_level=info</code></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description &amp; Example</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------</td>
</tr>
</tbody>
</table>
| –storage      | Path to node database file (defaults to node.db in the same folder)  
storage=node.db |
| –history_dir  | Path to folder where compressed (cut-through) history files are stored. Defaults to same folder.  
history_dir=. |
| –temp_dir     | Path to temp folder for compressed (cut-through) history files. Must be on the same volume as history_dir  
temp_dir=. |

**Using CPU mining is not recommended**

Beam uses Equihash mining algorithm with (150,5) parameters and customized data path. It is efficiently mined on GPUs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description &amp; Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>–miner_key</td>
<td>Secret key to attribute mining rewards mined by the node to your wallet Created using CLI wallel export_miner_key command with –subkey=&lt;miner id&gt; parameter See Command Line Wallet User Guide for more details</td>
</tr>
<tr>
<td>–owner_key</td>
<td>Secret key allowing the node to monitor mining rewards mined by all mining nodes marked by this key. Created using CLI wallel export_owner_key command See Command Line Wallet User Guide for more details</td>
</tr>
<tr>
<td>–pass</td>
<td>Wallet password. It is required since both Miner Key and Owner Key are protected by wallel password</td>
</tr>
</tbody>
</table>
| –stratum_port | Port on which stratum server will listen to incoming connections. 0 if stratum server is disabled.  
stratum_port=0 |
| –stratum_secrets_path | Path to folder containing stratum certificates  
stratum_secrets_path=. |

**Warning:** The following document is still under construction and is subject to changes

### 1.4.5 Backup and Restore

This document lists all backup and restore procedures necessary to keep your data safe.
**Restore Desktop wallet from Seed Phrase**

To be completed…

**Restore CLI wallet from Seed Phrase**

It is only possible to restore your coins by running your own node with your *owner key*.

This process involves several steps:

First you need to recreate your wallet using your *seed phrase* by running the following command:

```
./beam-wallet restore --seed_phrase=<semicolon separated list of 12 seed phrase words>
```

Now you need to export the owner key by running:

```
./beam-wallet export_owner_key
```

(see *Exporting owner key* for more details)

Then you need to run your own node, providing the owner key as a parameter as follows:

```
./beam-node --peer=<ip and port of peer node> --owner_key=<owner key exported from, the wallet>
```

Once the node has synchronized, you need to connect your wallet to the node to update the wallet database.

To do that run the following command:

```
./beam-wallet listen -n <ip and port of your node, ex:127.0.0.1:10000>
```

After wallet synchronizes, use `info` command to check wallet status

```
./beam-wallet info
```

**Warning:** The following document is still under construction and is subject to changes

---

**1.4.6 Mining Beam**

Alike most cryptocurrencies, Beam relies on miners to add transactions to the blockchain. While all nodes in the Beam network confirm the validity of transactions, Beam counts on miners to take on the massive heavy lifting to guard the network.

Beam is a Mimblewimble implementation. We use classic Proof-of-Work (PoW) consensus.

We welcome everyone to join our mining community to support the network and earn Beam coins.

**Mining Algorithm**

To secure the network, Beam uses the uses a modified version of Equihash (150,5) proof-of-work mining algorithm. Miners compete against each other using their computing power produce a new block on the chain. The first miner that gets to complete the precise computation for each block is granted with a network standard block reward and any fees for transactions added to that block.
At Mainnet launch, we will use the following Equihash parameters: $n=150$, $k=5$. In addition, we will introduce a small change to the datapath to further reduce the chance of zero-day ASICs.

The minimal memory requirement for the GPU will be 4 GB. The most up-to-date list of supported GPUs will be available here.

**Block Size and Time**

A Beam block will be generated approximately every minute and contain about 1000 transactions. Block size will be roughly 1MB.

**Mining Difficulty**

Mining difficulty is a measure of how many attempts on average it is required to find the proof-of-work solution required to mine a block and receive the mining reward. One can define the Difficulty as the inverse probability of a random solution being the correct one. Thus, a difficulty of 100 means that one in 100 tries should produce a valid block in average.

In Proof of Work blockchains, the difficulty is a dynamic parameter, periodically retargeted to reflect the fluctuations in the total computational power of all the miners. Accounting for changes in mining competition, as well as improvements in mining technology.

The goal of updating the difficulty is to keep the average block time at a certain value. In Beam, the target block time is 60 seconds, which underpins a constant currency issuance and the transaction settlement time of the network.

Here's an example. Let's assume we have 100 miners in our network. They are trying different random solutions and find one roughly every minute. Now 100 more miners join, the hashrate of the network doubles, and it will take just 30 seconds to find the solution in average. And if 200 and more join, the time will halve into 15 seconds, and so on.

To mitigate that, Beam nodes adjust [shall we mention that they reach consensus here?] the difficulty, increasing the difficulty of the proof of work algorithm as more miners join, and lowering it if the number of miners decreases.

In Beam, the difficulty is reassessed with every block, by every client independently. The algorithm looks at the average time and the average difficulty the last 1440 blocks. If the time required to mine the last 1440 blocks is higher or lower than 24 hours, the difficulty is retargeted accordingly.

The detailed algorithm is outlined below:

1. Look at the last 7 blocks. Identify the block that has the median block time. This is the Window End block.
2. Look at blocks from 127 to 120 before the current block (7 blocks altogether). Identify the block that has the median block time out of those. This will be the Window Start block.
3. Sum all the difficulties of the blocks from Window Start to Window End. This is Delta Work.
4. Calculate the time difference between Window Start and Window End blocks. This is Delta Time.
5. Calculate the difficulty for next block as: NewDifficulty = (Delta Work / Delta Time) * 60 seconds.
6. The Delta Time is bounded by 1 hour and 4 hours. I.e. if the difference is less than 1 hour, 1 hours is still used as Delta Time, same for differences larger than 4 hours. This is done to prevent extreme changes in difficulty.

The algorithm above ensures that as the total solution power of the network fluctuates, the difficulty is gradually adjusted and thus the target block time is ensured.
Miner Rewards

During the first year of Beam existence, miner reward will be 80 coins per block. In years 2-5 the reward will be 40 coins per block. In year 6 the reward will be 25 coins, and then halving will occur every 4 years until year 129. After year 133, Beam emission will stop.

Mining reward (coinbase UTXO) has 4 hours maturity, meaning that it will be available for spending 4 hours after it was mined.

Treasury

In the first five years of existence, additional coins will be issued to Beam Treasury with each newly mined Beam block.

In the first year, the Treasury will receive additional 20 Beams per block, and in the years 2-5 the Treasury will receive 10 coins per block.

The Treasury will be used to repay Beam investors, Incentivize the Core Team and to support the Beam Foundation (largest single beneficiary of the Treasury).

The distribution of the Treasury Coins is performed on a quarterly basis in the following proportion:

- Investors: 40%
- Core Team: 40%
- Beam Foundation: 20% (Biggest single beneficiary)

ASIC Resistance

To ensure better decentralization, Beam plans to stay ASIC resistance in the first 12-18 months. To achieve this, we plan to perform one or two hard forks – first after approximately 6 months of existence and another one after approximately 12 months. Each hard fork will change the mining algorithm. The exact modifications will be revealed several weeks before the actual hard fork.

Mining Guide

The following section describes how to set up mining for Beam Network

Mining using external miner

This is a step by step guide on how to setup mining using stand alone Beam Node with Stratum Server and a mining client.

Before you start with the steps, please review the sample Mining Architecture

Important points to understand

- Beam node should connect to some other node on the network via –peer parameter (for example –peer=3.0.115.1:8100 for testnet 4)
- Beam node should run Stratum server by setting up –stratum_port parameter (for example –stratum_port=10002)
- Beam node should know both mining key and owner key in order to attribute mining rewards to a specific wallet. Mining and owner keys are exported from the wallet as explained in the steps 5 and 6 of the step by step guide. They are passed to the node via –miner_key and –owner_key parameters respectively
Important: Miner keys and Owner key should be kept secret at all times

- Several Mining clients can connect to the same node. Mining clients should run on machines with GPUs.
- The connection between the Mining Client and the Beam Node is encrypted using TLS. In order for TLS to work you need to provide a certificate file and secret keys for it, as described in Step 9.
- Mining Client and Beam Node should use the same API key. Details on how to setup an API key are described in Step 9.
- Your wallet will see mining rewards only if the following conditions hold:
  1. It was created using the same seed phrase as the wallet which created the miner and owner keys
  2. It is connected to the node that knows the owner key.

Note: In case of Desktop wallet, you need to run local node from within the wallet (which automatically knows the owner key) or connect to are remote node that know the owner key. If you just connect to random node you will NOT see your mining rewards

Beam node should

Now let’s start with the steps:

1. Download CLI Wallet archive for your platform from Beam Website
2. Extract the CLI Wallet to any folder on your machine (we will call it Wallet Folder)
3. Open a Terminal window (on Mac / Linux) or Command Prompt (on Windows) and change directory to Wallet Folder

**Attention:** The following steps involve seed phrase and should be done in a secure environment to avoid someone stealing your seed phrase

4. If you want to create a new wallet (with new seed phrase) run the following command:

```
./beam-wallet init
```

If you have already created a wallet run the following command:

```
./beam-wallet restore --seed_phrase=<semicolon separated list of 12 seed phrase words>
```

**Attention:** Remember your wallet password, you will need it in step 11 to run Stratum server

5. Export miner key by running the following command

**Warning:** Mining key should be kept secret. Never send it or show it to anyone.

```
./beam-wallet export_miner_key --subkey=1
```

**Note:** If you want to run several different mining nodes with different keys you can run the command again with different subkeys.

**Example:** For second node, run: `./beam-wallet export_miner_key --subkey=2`

For third node, run: `./beam-wallet export_miner_key --subkey=3`

and so on

Save the exported mining key in some text file, you will need it later

6. Export owner key by running the following command:

**Warning:** Owner key should be kept secret. Never send it or show it to anyone.

```
./beam-wallet export_owner_key
```

Save the exported owner key in some text file, you will need it later

7. Download Beam Node archive for your platform from Beam Website

8. Extract Beam Node to any folder on your machine (from now on we will call it Node Folder)

9. Create certificate and API key for Stratum server

**Note:** If you are only testing you can download the sample certificate and key files from here:

Certificate File
Certificate Secret Key File

API Keys file

API Keys file currently contains one key: aaaa1234. You will need to provide it later in the --key=aaaa1234 parameter for miner client.

You should copy these files to the Beam Node folder (same folder as beam-node binary)

You can now jump to step 10

---

**Attention:** For production setup please read the following section carefully

Beam node implements Stratum protocol for connecting external miner clients. Clients open a TCP connection to the node through which they receive jobs to mine blocks using Equihash mining protocol.

Stratum server connections are protected using Transport Layer Security (TLS) protocol and require TLS certificates in order to work properly. You can either buy the certificates or create self signed certificates on your local machine. Instructions on how to do this are outside the scope of this guide. You should receive two files: one for certificate and one with the certificate secret key. For testing purposes you can always use sample files provided in the note above.

In addition you should create a file ‘stratum.api.keys’ which will contain one or more lines. Each line represents one API key - random strings of 8 characters or more. You should generate these keys yourself and put each one in new line. These keys are then used by the miner client via --key flag.

As a result you will have three files:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stratum.crt</td>
<td>TLS certificate</td>
</tr>
<tr>
<td>stratum.key</td>
<td>Private key for TLS certificate</td>
</tr>
<tr>
<td>stratum.api.keys</td>
<td>Text file with list of allowed API keys</td>
</tr>
<tr>
<td></td>
<td>Each key should have 8 symbols or more. example: abcd1234</td>
</tr>
</tbody>
</table>

All three files should be copied into the same folder. The path to this folder will be provided via --stratum_secrets_folder parameter. By default the path points to the same folder as the node binary.

At this point Node Folder should look something like this:

```
29/12/2018 07:43 <DIR> .
29/12/2018 07:43 <DIR> ..
28/12/2018 13:04 1,374 beam-node.cfg
28/12/2018 13:14 4,523,008 beam-node.exe
29/12/2018 07:41 2,670,592 node.db
28/12/2018 13:46 8 stratum.api.keys
28/12/2018 13:45 1,233 stratum.crt
28/12/2018 13:45 1,704 stratum.key

6 File(s) 7,197,919 bytes
2 Dir(s) 32,194,318,336 bytes free
```

10. Open a Terminal window (on Mac / Linux) or Command Prompt (on Windows) and change directory to Node Folder

11. Run Beam Node with stratum server using the following command:
./beam-node
      --port=10001
      --peer=3.0.115.1:8100
      --stratum_port=10002
      --stratum_secrets_path=.
      --miner_key=<mining key you got in step 5>
      --owner_key=<owner key you got in step 6>
      --pass=<your wallet password (not seed phrase)>

Note: Parameters in the example above are good for testing. You can always change them if necessary. You can also change the beam-node.cfg file and set all these parameters there instead of the command line.

The following table describes all parameters in more details

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description &amp; Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Port to start the server on</td>
</tr>
<tr>
<td>stratum_port</td>
<td>Port the stratum server is listening for incoming connec-</td>
</tr>
<tr>
<td>peer</td>
<td>Comma separated list of peer ip:port (must have at least</td>
</tr>
<tr>
<td>stratum_secrets_path</td>
<td>Path to a folder which holds TLS Certificate and API</td>
</tr>
<tr>
<td>miner_key</td>
<td>Miner key, exported by CLI wallet (see :ref: Creating</td>
</tr>
<tr>
<td>owner_key</td>
<td>Owner key, exported by CLI wallet</td>
</tr>
<tr>
<td>pass</td>
<td>Wallet password.</td>
</tr>
</tbody>
</table>

12. Downloads miner client archive for your GPU and platform from Beam Website

Beam provides two mining clients for Equihash 150.5 with data path change: one for OpenCL and one for CUDA
Attention: Only OpenCL mining client is currently available in Testnet

Note: Mining clients are only supported on Linux and Windows platforms

13. Extract miner client to a folder on your mining machine (from now on we wil call it Miner Folder)
14. Open a Terminal window (on Mac / Linux) or Command Prompt (on Windows) and change directory to Miner Folder
15. Run the following command (example on Windows):

```
beamMiner.exe --server 127.0.0.1:10002 --key aaaa1234
```

If your node runs on different machine than the miner, change IP address above to the IP of the node machine
If you have set a different API key than ‘aaa1234’ from the example set your key in the --key parameter.
Detailed explanation about mining client parameters is provided in the table below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description &amp; Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>IP and port of the Stratum server to connect to --server 127.0.0.1:10001</td>
</tr>
<tr>
<td>key</td>
<td>API key you have set in your Stratum server (In stratum.api.keys file) --key abcd1234</td>
</tr>
<tr>
<td>devices</td>
<td>Only specify this flag to use specific GPU By default, miner will use all available GPUs --devices 0</td>
</tr>
</tbody>
</table>

Your mining should start now.
To see your mining rewards use one of two options below:
1. Run Beam Desktop Wallet with the same seed phrase using built in node.
2. Run either CLI or Desktop wallet and connect it to Your node which was started with your owner key parameter (via –owner_key flag). It could be the same node as the miner, or another node - as long as it has your owner key

Warning: You will NOT be able to see your mining rewards if you connect to a node which does not know your owner key.

GPU Support
Here are some performance stats reported by our community
OpenCL Miner
<table>
<thead>
<tr>
<th>GPU</th>
<th>Supported</th>
<th>Reported Sol/s rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD RX560</td>
<td>Yes</td>
<td>~4</td>
</tr>
<tr>
<td>AMD RX570</td>
<td>Yes</td>
<td>~7-8</td>
</tr>
<tr>
<td>AMD RX580</td>
<td>Yes</td>
<td>~8-9</td>
</tr>
<tr>
<td>AMD Rx Vega 56</td>
<td>Yes</td>
<td>~13</td>
</tr>
<tr>
<td>nVidia GTX 1066</td>
<td>Yes</td>
<td>~5.25</td>
</tr>
<tr>
<td>nVidia GTX 1050Ti</td>
<td>Yes</td>
<td>~2.2-4.8</td>
</tr>
<tr>
<td>nVidia GTX 1060 6Gb</td>
<td>Yes</td>
<td>~5</td>
</tr>
<tr>
<td>nVidia GTX 1070</td>
<td>Yes</td>
<td>~7</td>
</tr>
<tr>
<td>nVidia GTX 1080</td>
<td>Yes</td>
<td>~8-9</td>
</tr>
<tr>
<td>nVidia GTX 1080Ti</td>
<td>Yes</td>
<td>~10-11</td>
</tr>
<tr>
<td>nVidia GTX 2080</td>
<td>Yes</td>
<td>~10-11</td>
</tr>
</tbody>
</table>

CUDA Miner

**Note:** CUDA Mining client is still in development.

### 1.4.7 Atomic Swaps using Desktop Wallet

**Note:** Atomic Swaps is an advanced feature, currently in beta.

DO NOT USE SWAPS FOR LARGE AMOUNTS OF MONEY

For Electrum we support only Legacy type seeds and SSL connection

For Bitcoin use version Bitcoin Core v0.17.1.

For Litecoin use version Litecoin v0.17.1.

**Introduction to Atomic Swap**

The idea of atomic swap was presented as solution for the next problem: two parties (Alice and Bob) want to exchange their coins without having to trust third party.

In a simplified way non-atomic approach consists of four steps:

1. Alice send her A Coins to Bob.
3. Bob sends his B Coins to Alice.
4. Alice receives B Coins.

The weakest part of this such approach is lurking in step #3. By malicious intent, Bob can avoid following his part of the agreement. Atomic swap is provided to guarantee compliance of the agreement.

The way this really works in practice is as follows:

1. Alice and Bob create three transactions between their wallets: Lock Transaction, Redeem transaction and Refund transaction.
2. Both parties lock their coins on the respective chain using the Lock Tx. In this state the coins belong to both parties and can not be spent.
3. Once the lock is confirmed, the parties exchange secret used to lock funds and send the Redeem Tx to the chain, effectively getting the swapped coins.

4. If the swap fails for any reason, Refund Tx can be sent to the network after some significant period of time to get the locked coin back for each party.

Configuring swaps in Desktop wallet

To enable swap functionality Beam wallet should first connect to the node and wallet of the other currency. This can be done using either your own instance of a full node via RPC or by using an Electrum service.

Connecting to BTC / LTC / QTUM full node

To connect to your own node perform the following steps (Litecoin example shown, for additional information and settings please use original documentation for Bitcoin, Litecoin and QTUM respectively)

1. Configure the litecoind to run with JSON RPC using the following commands in litecoin.conf configuration file

   ```
   # server=1 tells litecoin-QT to accept JSON-RPC commands.
   server=1
   
   rpcport=9432
   
   # You must set rpcuser and rpcpassword to secure the JSON-RPC api
   rpcuser=liteuser
   rpcpassword=123
   ```

   Make sure the node is running and fully synchronised

   2. Connect Beam Wallet to the node

   Open Settings tab in the Beam Wallet and open the Swap section using the ‘Swap’ toggle in the top right corner of the screen

   In the Litecoin section, fill in the node details, including ip:port of the node and the RPC username and password. The default fee parameter specifies the fee paid on the respective chain per Kb of transaction size and can be left at the default value.

   Once the settings for the node are specified click ‘Apply’ and than ‘Connect’

   If the connection is successful the green light will be shown near the section title

Connecting using Electrum Wallet

Another, simpler option to connect to a remote node would be using Electrum wallet service. First you need to find an address of an Electrum node, or run your own.

To connect to Electrum you need to turn on the ‘Electrum’ toggle in the appropriate section and specify the required parameters.

At this point you need to either enter an existing seed phrase for the wallet or create a new one.

You can later edit or create new wallet at any time

Once the seed is set you can connect to the Electrum wallet

Once the wallet is connected, you should be able to see the list of wallet addresses by clicking ‘Show wallet addresses’.

After you have completed the configuration, you are able to accept swap offers from Atomic Swaps tab
Settings

- **BITCOIN**
  - Node: Electrum
  - Node Address: tr.not.fyi:55002
  - Default fee: 90,000 sat/KB

- **LITECOIN NODE**
  - Node: Electrum
  - Disconnect
  - Node Address: 127.0.0.1:9432
  - Username: liteuser
  - Password: masked
  - Default fee: 90,000

- **QTUM**
  - Node: Electrum
  - Node Address
  - Username
  - Password
  - Default fee: 500,000

Buttons:
- Connect
- Cancel
- Apply
1.4. Reporting Issues and Getting Support
Before we discuss how to create and accept offers we need to understand the concepts of balance and UTXO.

**Understanding balances and UTXOs**

Since Beam, Bitcoin, Litecoin and QTUM are all UTXO based cryptocurrencies, understanding the difference between balance and UTXO is crucial for correct operation of the swap (and also regular) transactions.

Let’s say that you see available balance of 200 BEAM in your wallet and decide to swap half of it for a matching amount of Bitcoin. Once you initiate the swap, amount of free coins left in your wallet depends on the UTXOs that you had in the first place before the transaction started.

For example, if you had one UTXO of 200 BEAM, you will have 0 BEAM left until the swap is completed (even though you are only actually swapping 100). If you had two UTXO, 100 each, then you will have exactly

**Creating swap offer**

Click on ‘Create offer’ button in the top right corner of the swaps screen to create a new offer.

![Beam Wallet Testnet](image)
You can specify what amount and of which currency you want to trade to which amount of another currency. If you want to use the rate instead of explicitly setting the received amount, you can use the rate selector.

Once you have specified the swap details, you can either copy the swap token and send it to the specific opposite party using secure communication channel, alternatively you can publish the swap offer in the wallet so that it can be accepted by anyone.

If you choose to publish the offer you can cancel it at any time before it was accepted by clicking Cancel on the offer. To view only your offers, click ‘Only my offers’ checkbox.

**Accepting swap offer**

An Active offers table lists all currently offered swaps. The ‘Send’ column indicates what amount of which coin you will send in the swap and the ‘Receive’ column specifies what you will receive in return.

You can select the coin you want to swap in the dropdown list in the top right part of the list, and then switch ‘Send or receive BEAM’ toggle to see the matching offers.

Once you have seen the offer you like, and provided you have enough funds for the swap you can click ‘Accept’ button near the offer to review the swap details.
Click ‘Swap’ to accept the conditions and to initiate the swap process.

**Understanding swap transactions**

Every time a swap offer is created or accepted a matching swap transaction is created as well. The list of current swap transactions can be seen in the ‘Transactions’ tab of the Atomic Swaps screen.

![Atomic Swaps user interface](image)

Here you can see the current status of the transaction, and by clicking any transaction in the list, you can see its details. In case it is still possible to cancel the transaction you can do so by clicking ‘Cancel’ from the transaction menu (three vertical dots at the end of each line). Once transaction has completed (or expired or failed) you can delete the transaction from the list using the same menu.

### 1.4.8 Atomic Swaps using Command Line Wallet

**Note:** Atomic Swaps is an advanced feature, currently in beta.
DO NOT USE SWAPS FOR LARGE AMOUNTS OF MONEY
For Electrum we support only Legacy type seeds and SSL connection
For Bitcoin use version Bitcoin Core v0.17.1.
For Litecoin use version Litecoin v0.17.1.

Performing atomic swap with Litecoin

To perform atomic swap between Beam and Litecoin, Alice (who has BEAM) and Bob (who has LTC) need to follow the steps below:

1. Alice and Bob need to start full nodes for Beam and Litecoin blockchains.

Litecoin node should be configured to allow RPC access using either command line or config file as described in the Litecoin documentation.

In order to run Litecoin node Alice and Bob are using following command:

```
$ ./litecoind -server -datadir="path_to_litecoin_wallet_data" -rpcuser=<litecoin_RPC_username> -rpcpassword=<password> -printtoconsole
```

Alice example:

```
$ ./litecoind -server -datadir="Alice/path_to_litecoin_wallet_data" -rpcuser=Alice -rpcpassword=123 -printtoconsole
```

Bob example:

```
$ ./litecoind -server -datadir="Bob/path_to_litecoin_wallet_data" -rpcuser=Bob -rpcpassword=123 -printtoconsole
```

In this example we are using standard node and RPC ports.

**Attention:** The nodes should be sync up to current blockchain height before any swap operations.

In order to set up Beam node and command line wallet, please follow instructions *Command Line Wallet User Guide*

2. You can check your Litecoin balance using the following commands:

```
$ ./litecoin-cli -rpcuser=<rpc_user_name> -rpcpassword=<password> getbalance
```

Alice example:

```
$ ./litecoin-cli -rpcuser=Alice -rpcpassword=123 getbalance
```

Bob example:

```
$ ./litecoin-cli -rpcuser=Bob -rpcpassword=123 getbalance
```

3. Initialize swap settings:

```
$ ./beam-wallet set_swap_settings --swap_coin=ltc --swap_wallet_addr=<litecoin_node_ip:rpc_port> --swap_wallet_user=<litecoin RPC username> --swap_wallet_pass=<litecoin RPC password> --swap_feerate=<Litecoin fee rate(Photons/Kb)> --active_connection=core
```

1.4. Reporting Issues and Getting Support
Alice example:

$ ./beam-wallet set_swap_settings --swap_coin=ltc --swap_wallet_addr=127.0.0.1:13300 --swap_wallet_user=Alice --swap_wallet_pass=123 --swap_feerate=90000 --active_connection=core

Bob example:

$ ./beam-wallet set_swap_settings --swap_coin=ltc --swap_wallet_addr=127.0.0.1:13400 --swap_wallet_user=Bob --swap_wallet_pass=123 --swap_feerate=90000 --active_connection=core

**Attention:** Each coin has its own transaction feerate (\(--\text{swap\_feerate}\). To avoid failure or transaction jamming due to inconsistent fee amount, it’s recommended to check appropriate fee amount for each coin, and set it as \(--\text{swap\_feerate}\) value.

LTC \(--\text{swap\_feerate}\) = fee per 1 kb transaction size. Unlike Litecoin, Beam transaction feerate is static and doesn’t depend on transaction size. LTC value for swap amount (\(--\text{swap\_amount}\)) is provided in “photon”, 1 LTC = 1000000 photons, BEAM value is provided in beams.

4. It doesn’t matter who will run the swap first. In current case Bob starts:

$ ./beam-wallet swap_init -n <beam node ip:port> --amount=<amount of Beam> --swap_coin=ltc --swap_amount=<amount of Photons> --swap_beam_side

**Attention:** \(--\text{swap\_beam\_side}\) flag is used to point out a party changing BEAM to LTC (in this case Alice).

**Note:** The result of the swap_init command is a swap token.

Example:

$ ./beam-wallet swap_init -n "eu-node01.mainnet.beam.mw:8100" --amount=10 --swap_coin=ltc --swap_amount=2000000000

Bob copies swap token from console and send it to Alice. After that Bob should run wallet with listen command.

Example:

$ ./beam-wallet listen -n "eu-node01.mainnet.beam.mw:8100"

Alice will use next command to participate in the swap:

$ ./beam-wallet swap_accept -n <beam node ip:port> --swap_token=<Bob’s swap token>

Example:

$ ./beam-wallet swap_accept -n "eu-node01.mainnet.beam.mw:8100" --swap_token=316sveQtJrhxzUBy2zJHTp8aHfpdz2FycrR8n9fs5ChXgoq1Be4Z9gEPnz5HjxuB2gmQpxWd8Dy9icQVKvle23c7x5

5. If Alice accepts swap conditions, a swap transaction will be created, and LTC UTXO will be locked on Bob’s blockchain.

6. Alice and Bob need to wait for 6 blocks confirmation (in Litecoin blockchain) till the coins will be locked.
Note: Each blockchain has its own block generation time. For Litecoin average time equals 2.5 minutes, for Beam: 1 minute, for Bitcoin: 10 minutes.

7. After 6 blocks confirmation, Bob will redeem Alice’s beams and will reveal secret. After that the swap will be completed for Bob.
8. Alice will redeem Bob’s litecoins using secret. After that the swap will be completed for Alice.
9. Now Alice and Bob can check their Litecoin and Beam wallets accordingly to ensure the coins were transferred to them.

Performing atomic swap with Bitcoin

To perform atomic swap between Beam and Bitcoin, Alice (who has BEAM) and Bob (who has BTC) need to follow the steps below:

1. Alice and Bob need to start full nodes for Beam and Bitcoin blockchains.

   Bitcoin node should be configured to allow RPC access using either command line or config file as described in the documentation here.

   In order to run Bitcoin node Alice and Bob are using following command:

   For Alice:
   $ ./bitcoind -server -datadir="Alice/path_to_litecoin_wallet_data" -rpcuser=Alice -
   → rpcpassword=123 -printtoconsole

   For Bob:
   $ ./bitcoind -server -datadir="Bob/path_to_litecoin_wallet_data" -rpcuser=Bob -
   → rpcpassword=123 -printtoconsole

   In this example we are using standard node and RPC ports.

   Attention: The nodes should be synce up to current blockchain height before any swap operations.

   In order to set up Beam node and command line wallet, please follow instructions Command Line Wallet User Guide

   2. You can check your Bitcoin balance using the following commands:

      For Alice
      $ ./bitcoin-cli -rpcuser=Alice -rpcpassword=123 getbalance

      For Bob
      $ ./bitcoin-cli -rpcuser=Bob -rpcpassword=123 getbalance

   3. Initialize swap settings:

      $ ./beam-wallet set_swap_settings --swap_coin=btc --swap_wallet_addr=<bitcoin node> -
      → ip:rpc_port> --swap_wallet_user=<bitcoin RPC username> --swap_wallet_pass=<bitcoin_
      → RPC password> --swap_feerate=<Bitcoin fee rate(Satoshis/Kb)> --active_connection=core

      Alice example:
Bob example:

```bash
$ ./beam-wallet set_swap_settings --swap_coin=ltc --swap_wallet_addr=127.0.0.1:13400 -
\  \-swap_wallet_user=Bob --swap_wallet_pass=123 --swap_feerate=90000 --active_ 
\  \-connection=core
```

**Attention:** Each coin has its own transaction feerate (`--swap_feerate`). To avoid failure or transaction jamming due to inconsistent fee amount, it’s recommended to check appropriate fee amount for each coin, and set it as `--swap_feerate` value.

**BTC** `--swap_feerate` = fee per 1 kb transaction size. Unlike Bitcoin, Beam transaction feerate is static and doesn’t depend on transaction size. BTC value for swap amount (`--swap_amount`) is provided in “satoshi”, 1 BTC = 1000000 satoshis, BEAM value is provided in beams.

4. It doesn’t matter who will run the swap first. In current case Bob starts:

```bash
$ ./beam-wallet swap_init -n <beam node ip:port> --amount=<amount of Beam> --swap_ 
\  \-coin=btc --swap_amount=<amount of Satoshi> --swap_beam_side
```

**Attention:** `--swap_beam_side` flag is used to point out a party changing BEAM to BTC (in this case Alice).

**Note:** The result of the swap_init command is a swap token.

Example:

```bash
$ ./beam-wallet swap_init -n "eu-node01.mainnet.beam.mw:8100" --amount=10 --swap_ 
\  \-coin=btc --swap_amount=100000000
```

Bob copies swap token from console and send it to Alice. After that Bob should run wallet with listen command. Example:

```bash
$ ./beam-wallet listen -n "eu-node01.mainnet.beam.mw:8100"
```

Alice will use next command to participate in the swap:

```bash
$ ./beam-wallet swap_accept -n <beam node ip:port> --swap_token=<Bob's swap token>
```

Example:

```bash
$ ./beam-wallet swap_accept -n "eu-node01.mainnet.beam.mw:8100" --swap_ 
\  \-token=316w4oB5hCz2qeVNrteAEZXhxxx2HBX9v1Ped1FhveJor5JbChz2xXGfi2LkKqVLu8kU4vEo2CV3UbswoBZXBABJzmbx
```

5. If Alice accepts swap conditions, a swap transaction will be created, and BTC UTXO will be locked on Bob’s blockchain.

6. Alice and Bob need to wait for 6 blocks confirmation (in Bitcoin blockchain) till the coins will be locked.
Note: Each blockchain has its own block generation time. For Bitcoin average time equals 2.5 minutes, for Beam: 1 minute, for Bitcoin: 10 minutes.

7. After 6 blocks confirmation, Bob will redeem Alice’s beams and will reveal secret. After that the swap will be completed for Bob.

8. Alice will redeem Bob’s bitcoins using secret. After that the swap will be completed for Alice.

9. Now Alice and Bob can check their Bitcoin and Beam wallets accordingly to ensure the coins were transferred to them.

1.4.9 Blockchain Explorer

Note: Blockchain Explorer is currently running at https://explorer.beam.mw

Main Screen

In the top left corner there is a selector to switch between Mainnet and Testnet explorers.

In the top right corner the is a search field that allows searching for blocks by height, hash or a kernel id of the transaction within the block.

The two charts on the main page shows two charts: the current average block creation time (red line) and current block difficulty (green line) on the left and the average fee (blue line) on the right.

Under the charts there are panes presenting the current status of the blockchain:

Blockchain height - the current height of the blockchain
Latest block - creation time of the last block
Latest block difficulty - the difficulty of the last block

Coins in circulation (mined) - how many Beams were mined so far

Coins in circulation (treasury) - how many Treasury coins are currently in circulation

Total coins in circulation - Total amount of both mined and treasury coins

Next treasury emission block height - Block height of the next treasury burst

Next treasury emission coin amount - Amount of Treasury coins that will be released in the next burst

Total emission - maximum amount of Beam that will ever exist set to 262,800,000

Hashrate - current hashrate of the system (in Sol / sec) calculated as difficulty / 60

### Blocks

The list of most recent blocks is located at the bottom of the main screen

<table>
<thead>
<tr>
<th>Height</th>
<th>Hash</th>
<th>Age</th>
<th>Difficulty</th>
<th># inputs</th>
<th># outputs</th>
<th># freeze</th>
</tr>
</thead>
<tbody>
<tr>
<td>30062</td>
<td>9ea4f12bc3704c420543d9e51df9ad0f2b1b4319ecad65d7f3b569f5014</td>
<td>Jan 30, 2019, 12:31:02 PM</td>
<td>305,397,690</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30061</td>
<td>f13a3a975a51e68544c8c914d91be8e83797b2eb9b2d0317ca4e373c76060f</td>
<td>Jan 30, 2019, 12:30:49 PM</td>
<td>311,481,120</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30060</td>
<td>c5ebe166822029643110d8719a087f71af99de4682c0c680722637199</td>
<td>Jan 30, 2019, 12:30:27 PM</td>
<td>321,694,432</td>
<td>5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>30059</td>
<td>c451a95925e599fcd35e59cc2e6a69b76360476e8b75a370e569d47508c2598</td>
<td>Jan 30, 2019, 12:29:59 PM</td>
<td>321,678,330</td>
<td>5</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>30058</td>
<td>df51a2353336bc41a0e520c8d60552b071067c735679a114777ba9aa7c7e2c8a</td>
<td>Jan 30, 2019, 12:29:11 PM</td>
<td>330,500,015</td>
<td>50</td>
<td>70</td>
<td>20</td>
</tr>
<tr>
<td>30057</td>
<td>a1f0527139f38164e5d122550d73db97ba8f312503f03003c8e39e3w8f839f</td>
<td>Jan 30, 2019, 12:24:54 PM</td>
<td>303,300,020</td>
<td>37</td>
<td>09</td>
<td>20</td>
</tr>
<tr>
<td>30056</td>
<td>0be44a9eb626570f50b3ed9393c4fcd5d7e4c3d9ebf8b0852026650b4c14a3dd</td>
<td>Jan 30, 2019, 12:20:34 PM</td>
<td>300,405,005</td>
<td>60</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30055</td>
<td>a1f0527139f38164e5d122550d73db97ba8f312503f03003c8e39e3w8f839f</td>
<td>Jan 30, 2019, 12:20:29 PM</td>
<td>333,154,240</td>
<td>45</td>
<td>91</td>
<td>26</td>
</tr>
<tr>
<td>30054</td>
<td>0c21b2f6b105bccc43233445935be8e6b5876b70a409f97c958060e2e6c</td>
<td>Jan 30, 2019, 12:15:38 PM</td>
<td>331,926,995</td>
<td>60</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>30053</td>
<td>0c21b2f6b105bccc43233445935be8e6b5876b70a409f97c958060e2e6c</td>
<td>Jan 30, 2019, 12:15:27 PM</td>
<td>332,268,400</td>
<td>79</td>
<td>74</td>
<td>21</td>
</tr>
<tr>
<td>30052</td>
<td>0c21b2f6b105bccc43233445935be8e6b5876b70a409f97c958060e2e6c</td>
<td>Jan 30, 2019, 12:13:00 PM</td>
<td>332,709,296</td>
<td>6</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>30051</td>
<td>4e79f019a337f71a1e354e50803986223e7903846e9b2001402808f790</td>
<td>Jan 30, 2019, 12:12:29 PM</td>
<td>332,064,975</td>
<td>34</td>
<td>34</td>
<td>5</td>
</tr>
<tr>
<td>30050</td>
<td>0b593c2f08701b30bc32be3258b89455724204c6d530b30cdcdfe0552153c5b4a9</td>
<td>Jan 30, 2019, 12:11:12 PM</td>
<td>330,080,192</td>
<td>7</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>30049</td>
<td>2d58d4a4bcc6d4a74f5d9d560450655a9e43a0665abc26a9d10f</td>
<td>Jan 30, 2019, 12:10:43 PM</td>
<td>329,236,640</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Clicking on the hash link or the details triangle button on the right allows you to view details of each block

### 1.4.10 Supported Platforms

Beam Node and Beam Wallet are currently supported on the following platforms:

- **64-bit** Linux OS Ubuntu 14.04 LTS Trusty Tahr, Ubuntu 16.04 LTS Xenial Xerus, Ubuntu 18.04 LTS Bionic Beave
- **64-bit** Processor
- 5GB of free RAM
- 10GB of free Disk (*the size of the block chain increases over time*)

- **64-bit** Mac OS 10.13 or higher
- **64-bit** Processor
- 5GB of free RAM
- 10GB of free Disk (*the size of the block chain increases over time*)
Attention: At the moment Beam only works on processors that support AVX instruction set.

Note: Please report any issues you might encounter while running BEAM software on your system.

1.4.11 Files and Locations

Desktop Wallet app

General points to mention:

- The default location of the Desktop Wallet app can be modified during the installation process (Windows only).
- The default Database location for the Desktop Wallet app can be changed setting the appdata parameter to beam-wallet.cfg (Windows only).
- Memory dump files are generated on Windows only. A dedicated memory dump file is created per each crash case.
- Each version of the wallet keeps the wallet database file (wallet.db) in the dedicated sub-folder, designated by the version number. On each wallet update the new folder is created to which and the wallet database file from previous version is copied into (and updated if necessary).
Windows

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Executable</td>
<td>\Program Files\Beam&lt;version&gt;\Beam Wallet.exe</td>
</tr>
<tr>
<td>Configuration</td>
<td>\Program Files\Beam&lt;version&gt;\beam-wallet.cfg</td>
</tr>
<tr>
<td>Logs</td>
<td>\Users{your User name}\AppData\Local\Beam Wallet\logs</td>
</tr>
<tr>
<td>Database</td>
<td>\Users{your User name}\AppData\Local\Beam Wallet&lt;version&gt;\wallet.db (node.db)</td>
</tr>
<tr>
<td>Dumps</td>
<td>\Users{your User name}\AppData\Local\Beam Wallet\Beam Wallet.exe0.dmp</td>
</tr>
</tbody>
</table>

Mac

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Executable</td>
<td>/Applications/Beam Wallet.app</td>
</tr>
<tr>
<td>Configuration</td>
<td>N/A</td>
</tr>
<tr>
<td>Logs</td>
<td>/Users/[your User name]/Library/Application Support/Beam Wallet/logs</td>
</tr>
<tr>
<td>Database</td>
<td>/Users/[your User name]/Library/Application Support/Beam Wallet/&lt;version&gt;/wallet.db</td>
</tr>
</tbody>
</table>

Linux

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Executable</td>
<td>/usr/bin/BeamWallet</td>
</tr>
<tr>
<td>Configuration</td>
<td>/usr/bin/beam-wallet.cfg</td>
</tr>
<tr>
<td>Logs</td>
<td>/home/[your User name]/.local/share/Beam Wallet/logs</td>
</tr>
<tr>
<td>Database</td>
<td>/home/[your User name]/.local/share/Beam Wallet/&lt;version&gt;/wallet.db (node.db)</td>
</tr>
</tbody>
</table>

Node or CLI wallet

All Platforms (small differences apply, see below)

User can unpack the archive in any folder to his convenience. All files mentioned below are located within this folder

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Executable</td>
<td>beam-node or beam-wallet</td>
</tr>
<tr>
<td>Configuration</td>
<td>beam-node.cfg or beam-wallet.cfg</td>
</tr>
<tr>
<td>Logs</td>
<td>logs</td>
</tr>
<tr>
<td>Database</td>
<td>node.db or wallet.db</td>
</tr>
<tr>
<td>Dumps (windows only)</td>
<td>beam-node.exe0.dmp</td>
</tr>
</tbody>
</table>

1.4.12 How to run Testnet and Mainnet on the same machine

The following networks exist in Beam realm: * Mainnet: the network with real money and actual transactions. * Testnet: staging environments for trying new features and helping to find the early bugs in our wallet, node, and miner software. For advanced users only. * Masternet: new features in the daily development cycle, if you find yourself on this network it means that you are either very early adopter or Beam code contributor.

Hence, sometimes you wanna play with the new features on Testnet while sending the “real money” over the Mainnet. It’s easy as most features work out-of-the-box and slight visual differentiations such as backgrounds or special
messages will always hint whether you are on Mainnet or Testnet. In case you’d like to get a bit deeper into the tech
details, here’s what you need to know:

**Downloading binaries**

Mainnet address is https://www.beam.mw/downloads/mainnet, Testnet address is https://www.beam.mw/downloads/
testnet Testnet binaries will always include “testnet” in the filename, e.g. Beam-Wallet-Testnet-x.y.zzz.dmg

**Desktop wallet app and CLI wallet**

- In case desktop wallets run integrated nodes, make sure that they use different ports. By default, the Mainnet
  node will use 10005 port, while Testnet will use 11005 port.
- For desktop wallet, the port of the integrated node can be set in the Settings screen.
- Mainnet wallet files and logs are kept in the Beam Wallet folder, Testnet wallet files are in Beam Wallet Testnet
  folder by default.
- Desktop wallet and download pages on Testnet will have a dark violet background, Mainnet background will
  remain the same.

**Node**

- Mainnet node will use 10000 port, while Testnet will use 11000 port by default.
- Mainnet node will reject a connection request from a Testnet wallet, same applies for the Testnet node, rejecting
  a connection from a Mainnet wallet.

**Miner**

- Currently, the miner is “just” a pure PoW calculation software. Hence, any Beam miner can work with either
  Mainnet or Testnet node. When configuring the miner-to-node connection, make sure your miner is connected
to the network as intended.

### 1.4.13 Reporting Issues and Getting Support

To report an issue please use our GitHub.

To get support please contact our Technical Support Team via:

1. Telegram
2. Discord
3. Email (support@beam-mw.com)

### 1.4.14 Wallet Troubleshooting (all wallets)

**Where are the wallet files located?**

When Beam Wallet desktop app is installed, the wallet data files are stored separately from the binaries. The locations
of all the files are described here: Files and Locations
My transaction is stuck ‘In Progress’ for a long time

In progress means that the message sent to the other wallet address was not answered yet. Each message has a lifetime of 12 hours, so if the message will not be answered during that time transaction will be canceled automatically by the wallet. At this stage, the sender can cancel the transaction by clicking on transaction menu and selecting ‘Cancel’.

My transaction is stuck in ‘Synching with blockchain’ for a very long time

In order to create a transaction Sender and Receiver should exchange messages with all the necessary information. After that, Sender creates the transaction and sends it to the network for distribution and mining.

‘Synching with blockchain’ is a state in which wallet waits for a message from the other side. For a ‘Sender’ it means that the message was sent and not answered yet. For a ‘Receiver’ it means that the answer was sent but the transaction is not yet visible in the blockchain either because it was not sent to the network or because it was not mined yet.

In any case, if the transaction does not appear in the blockchain after 2 hours it is automatically canceled by the wallet.

I’ve forgot the local password for my wallet

See :ref: Restoring funds

I’ve restored the wallet but my balance is zero

One or more the of the words is wrong or misspelled. Triple-check that all the words from the seed phrase are typed in correctly. You will need to repeat the :ref: Restoring funds procedure.

I’ve forgot my password

If you lost your password and cannot get into your wallet, you will have to remove wallet.db file and to Restore funds using your seed phrase to create a new password.

Why is the seed phrase the only thing connecting me to my funds?

To ensure the utmost privacy, the only information we can use to link you to your wallet is your seed phrase. So, if you lose it we cannot recover it for you.

I’ve lost my seed phrase

By design, the only way to access your funds (UTXO) is to have the seed phrase. If you still have access to your wallet, create another wallet with new seed phrase on another device and transfer funds to there. Any solution that would allow you to access your funds without the seed phrase would severely compromise the privacy of BEAM. Therefore, in case you don’t have any active access to your funds there is nothing to do (the funds will be stored in the blockchain forever and no one will be able to access or spend them).

I’ve copied my wallet.db file to the new machine and I’d like to run wallets on both new and old machines simultaneously

At the current implementation each wallet.db file should be managed by only a single wallet instance. Any case involving manual transfer of the wallet database is not supported.
I am trying to send Beam but transactions are failing

In certain cases, the wallet may get out of sync with the blockchain which might result in UTXOs that were already spent being incorrectly marked as available. When such UTXOs are selected for a transaction by the wallet, the transaction will be rejected by the blockchain.

To fix the situation, do the following:

1. Open the Wallet and open Settings tab
2. Switch to a local node
3. Click on ‘Rescan’ button
4. Wait for the wallet to synchronize

In some cases this operation may result in change if your wallet balance, which was incorrectly displaying already spent UTXOs as available.

If this does not help, you may try to resync the wallet completely by following the procedure below:

1. Erase the wallet.db (you can back it up), because it may continue to create duplicated coins. Then restore it via your secret phrase. No need to erase node.db.
2. After the sync is complete - send all your “visible” funds to yourself. You can set fee=0. Wait until transaction completes.
3. Wait for 2 days. Meanwhile you may use your wallet normally, but some of the funds may still look missing.
4. After 2 days: Erase both wallet.db AND node.db. Then - do a full restore.

1.4.15 Desktop Wallet Troubleshooting

Why is my available balance lower than expected while I’m sending BEAM?

UTXO can be locked during active outgoing transaction. The locked amount is displayed as a change in ‘Sending screen’. The change will become spendable when the transaction expires or completes.

I’ve restored the wallet but I can’t see my transaction list and/or my active addresses

As explained in Restoring funds, only your available balance (i.e. your UTXO) is kept on the blockchain, hence that’s all that can be restored.

I’ve restored the wallet using my seed phrase - can someone still send me money to the addresses created in the previous wallet?

When a wallet is restored, only the balance (UTXO) is restored. Addresses (active and expired), contacts and transaction history are only stored locally, so they can’t be restored from the blockchain. Each wallet is aware of only the active and expired addresses it displays. Therefore, all transactions sent to the addresses no wallet is aware of anymore will fail by timeout and the funds will be automatically released in Sender’s wallet.

Why can’t I just cancel the transaction in the ‘Synching with blockchain’ state?

Your wallet has already disclosed enough information so that transaction can be created anyway and sent to the network even if you cancel it.
**Wallet is stuck in ‘Downloading blocks’ screen**

1. Close your wallet
2. Locate the Beam Wallet folder *Files and Locations*
3. Use any text editor to open settings.ini file
4. Check the contents of the ‘peers’ value

```
[localnode]
port=10005
run=true
peers=@Invalid()

[node]
address=us-node01.mainnet.beam.mw:8100
```

5. If the value is `@Invalid()` replace it with the following:

```
[localnode]
port=10005
run=true
peers=eu-node02.mainnet.beam.mw:8100, eu-node01.mainnet.beam.mw:8100, us-node02.
   mainnet.beam.mw:8100, us-node04.mainnet.beam.mw:8100, ap-node01.mainnet.beam.mw:8100,
   ap-node02.mainnet.beam.mw:8100

[node]
address=us-node01.mainnet.beam.mw:8100
```

**My peers look ok but the wallet is still stuck during sync**

1. Close your wallet
2. Locate the Beam Wallet folder *Files and Locations*
3. Delete node.db file and all files starting with ‘tempmb’
4. Restart the wallet

### 1.4.16 Command Line Wallet Troubleshooting

**I am getting the error code=26, file is not a database error when starting the command line wallet**

Notice how you were starting the wallet:

```
I 2018-12-23.17:32:34.619 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.17:32:34.620 starting a wallet...
Enter password: ***
D 2018-12-23.17:32:34.664 sqlite error code=26, file is not a database
E 2018-12-23.17:32:34.665 Wallet data unreadable, restore wallet.db from latest
   backup or delete it and reinitialize the wallet
```

You have submitted an incorrect password. The wallet can not decrypt the database file and hence reports that data is unreadable. *Only if you are absolutely sure that password is correct, remove the database file and restore wallet from your Seed Phrase*
I am getting the Failed. No inputs exception when starting the command line wallet

Notice how you were starting the wallet:

```
I 2018-12-23.17:45:12.529 Rules signature: ddccf5d8d0f77bd2
I 2018-12-23.17:45:12.530 starting a wallet...
Enter password: *
I 2018-12-23.17:45:13.226 wallet sucessfully opened...
I 2018-12-23.17:45:13.228 WalletID→14a38140d8e66be9b8f1e88d770161fd33e35f7000053147b5a0f6a83178926b956 subscribes to→BBS channel 20
I 2018-12-23.17:45:13.271 [9edc454f2752461eb682f21c4efbd33e] Sending 10 beams (fee:
→0 groth )
E 2018-12-23.17:45:13.272 You only have 0 groth
E 2018-12-23.17:45:13.273 [9edc454f2752461eb682f21c4efbd33e] exception msg:
I 2018-12-23.17:45:13.293 [9edc454f2752461eb682f21c4efbd33e] Transaction failed.→Rollback...
```

The most common cause of this error is trying to send a transaction with insufficient funds. You can not send a greater amount than you have.

### 1.4.17 Miscellaneous Troubleshooting

**My question is not answered anywhere**

See *Reporting issues and getting support*

---

**Warning:** The following document is still under construction and is subject to changes

### 1.4.18 Resources

This page includes a list of useful and recommended resources to learn about cryptocurrencies in general and Beam in particular.

- **Blockchain**
- **Cryptocurrencies**
- **Mimblewimble**
- **Beam**

### 1.4.19 Glossary

**Address**

In Beam, addresses are only used by SBBS system to connect between Wallets during transaction creation. Unlike most other blockchains, addresses are not recorded in the blockchain and are not used to prove ownership of the coins. Each address has a default expiration time of 24 hours (which can be changed using Wallet UI). In general, it is recommended to generate fresh receiving address for each transaction.
Blinding factor

Blinding factor is a secret key that is deterministically derived from a master key and is used to

**Block** A block is a record in the Beam blockchain that contains a set of transactions sent on the network. Pending inclusion in a block, a transaction is kept in the ‘mempool’ in an ‘unconfirmed’ state. Roughly every 2.5 minutes, on average, a new block is appended to the blockchain through ‘mining’ and the transactions included receive their first ‘confirmation’.

**Block reward** A block reward is new ‘beams’ created after the successful ‘mining’ of a block. For the first five years, the block reward in Beam is split into a miners’ reward and a treasury. During this time, miners receive 80% per block with the remaining 20% split between Beam Foundation, founders, employees, investors and advisors. Block reward is subjected to periodic halving and is halved for the first time after about one year (in blocks) and then every four years.

**Blockchain** The blockchain is a public record of Beam transactions. The blockchain can be downloaded and verified by all Beam nodes to make sure the transaction history is valid and no double spending occurs. In Beam, a compacted version of the blockchain is maintained by each node using the cut through feature of mimblewimble which enables new nodes to only download minimal amount of information to start mining and verifying new blocks.

**Blockchain Explorer** The Blockchain explorer is a website showing the current status of Beam blockchain and history of mined blocks.

**Cut through** In mimblewimble protocol, cut through feature enables creation of compacted, yet still verifiable, blockchain history by removing all intermediate transactions and thus significantly reducing the amount of information required by a new node in the system to start mining or verifying new blocks.

**Macroblock**

A macroblock is a compressed version of blockchain history implementing the cut-through feature of Mimblewimble protocol. Each node generates macroblocks in the background and stores them on the local disk. When new node connects to the system it first downloads the latest Macroblock and then updates more recent blocks in the blockchain one by one. This allows to significantly reduce the time of onboarding new nodes into the system.

**Master key** Master key (or master secret key) is the key used to generate all blinding factors in a single wallet. Master key is also used to generate Miner Keys and Owner key used to mine coins for a specific wallet. In general

**Mining difficulty**

Mining difficulty is a dynamic parameter which determines amount of calculations necessary to solve Beam Proof of Work puzzle. It is designed to ensure that blocks are created once a minute (on average) regardless of number of miners in the network. Current block creation time and difficulty can be seen using the blockchain explorer.

**Seed Phrase**

A list of 12 words that hold all information necessary to generate Master Key and hence to recover all Beam UTXOs belonging to a wallet created using this phrase. Seed phrase is generated during wallet initialization and should be kept secret at all times. If lost, the Seed Phrase CAN NOT BE RECOVERED by any means. Keep it safe.

**Mimblewimble** Mimblewimble is the protocol used by Beam to provide confidentiality of transactions and scalability in terms of compact blockchain size. Mimblewimble white paper was published in July 2016 by an anonymous author under the pseudonym Tom Elvis Jedusor.

SBBS
SBBS (abbreviation of Secure Bulletin Board System) is a subsystem within Beam Node that allows wallets to securely exchange encrypted messages and create transactions without having to be online at the same time.

**Wallet Password**

Wallet Password is a password that protects wallet and encrypts wallet database used to store information about UTXOs, transactions and any additional metadata stored by the wallet. Wallet Password is NOT used for or has any relation to ownership of the coins. If Wallet Password is lost, the wallet database can no longer be accessed and the transaction history and metadata will be lost. However it is possible to recover all the currently owned UTXOs, by creating a new wallet and initializing it using the same seed phrase as the original one.

**Transactions**

In mimblewimble protocol transactions contain of Inputs, Outputs and Kernels. Each input and output are represented by Pedersen Committments in a form: \( P = v*H + b*G \), where \( v \) is transaction value, \( b \) is the blinding factor and \( G \) and \( H \) are two known ‘nothing up my sleeve’ generator points on the same elliptic curve.

### 1.4.20 Building Beam

**Warning:** The following document is still under construction and is subject to changes

The following document describes how to build Beam binaries from sources located at: [https://github.com/BeamMW/beam](https://github.com/BeamMW/beam)

**Windows**

1. Install Visual Studio >= 2017 with CMake support.
2. Download and install Boost prebuilt binaries [https://sourceforge.net/projects/boost/files/boost-binaries/1.68.0/boost_1_68_0-msvc-14.1-64.exe](https://sourceforge.net/projects/boost/files/boost-binaries/1.68.0/boost_1_68_0-msvc-14.1-64.exe), also add BOOST_ROOT to the Environment Variables.
3. Download and install OpenSSL prebuilt binaries [https://slproweb.com/products/Win32OpenSSL.html](https://slproweb.com/products/Win32OpenSSL.html) (Win64 OpenSSL v1.1.0h for example) and add OPENSSL_ROOT_DIR to the Environment Variables.
4. Download and install QT 5.11 [https://download.qt.io/official_releases/qt/5.11.0/qt-opensource-windows-x86-5.11.0.exe.mirrorlist](https://download.qt.io/official_releases/qt/5.11.0/qt-opensource-windows-x86-5.11.0.exe.mirrorlist) and add QT5_ROOT_DIR to the Environment Variables (usually it looks like ‘/5.11.0/msvc2017_64’), also add QML_IMPORT_PATH (it should look like ‘%QT5_ROOT_DIR%qml’). BTW disabling system antivirus on Windows makes QT installing process much faster.
5. Add ‘/5.11.1/msvc2017_64/bin’ and ‘/boost_1_68_0/lib64-msvc-14.1’ to the System Path.
6. Open project folder in Visual Studio, select your target (Release-x64 for example, if you downloaded 64bit Boost and OpenSSL) and select CMake -> Build All.
7. Go to CMake -> Cache -> Open Cache Folder -> beam (you’ll find beam.exe in the beam subfolder, beam-wallet.exe in ui subfolder).

**Linux (Ubuntu 14.04 and higher)**

1. Install gcc7 boost ssl packages.
### 2. Set it up so the symbolic links gcc, g++ point to the newer version:

```bash
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 60
--slave /usr/bin/g++ g++ /usr/bin/g++-7
```

```
gcc --version
g++ --version
```

### 3. Install latest CMake

```bash
wget "https://cmake.org/files/v3.12/cmake-3.12.0-Linux-x86_64.sh"
sudo sh cmake-3.12.0-Linux-x86_64.sh --skip-license --prefix=/usr
```

### 4. Add proper QT 5.11 repository depending on your system [https://launchpad.net/~beineri](https://launchpad.net/~beineri) (for example, choose Qt 5.10.1 for /opt Trusty if you have Ubuntu 14.04), install `sudo apt-get install qt510declarative qt510svg` packages and add `export PATH=/opt/qt/bin:$PATH`.

### 5. Go to Beam project folder and call `cmake -DCMAKE_BUILD_TYPE=Release . && make -j4`.

### 6. You’ll find `Beam` binary in `bin` folder, `beam-wallet` in `ui` subfolder.

### Mac

1. Install Brew Package Manager.
2. Install necessary packages using `brew install openssl boost cmake qt5` command.
5. You’ll find `Beam` binary in `bin` folder, `beam-wallet` in `ui` subfolder.

**Note:** If you don’t want to build UI don’t install QT5 and add `-DBEAM_NO_QT_UI_WALLET=On` command line parameter when you are calling `cmake`.

**Warning:** The following document is still under construction and is subject to changes.

### 1.4.21 Understanding Beam logs

#### Log locations

For CLI wallet and Node logs are usually located in the same folder as the binary file.

For Desktop Wallet logs are located in the following folders:

- **Mac:** `/Users/{your_user_name}/Library/Application Support/Beam Wallet/`
- **Windows:** `\Users\{your_user_name}\AppData\Local\Beam Wallet`
Linux: /home/{your_user_name}/.local/share/Beam Wallet
For a complete list of file locations see :ref: Files and Locations

**Node logs**

A log will start with the Rules signature. Rules signature is the hash of the Consensus Rules and it should be compatible with the network you are connecting to. It will be different between the Testnet and Mainnet. The relevant Rules signature for each network can be seen here: :ref: rules_signature

Beam Logs have a simple structure. First field is the severity level, followed by a timestamp and the log message.

<table>
<thead>
<tr>
<th>Log Severity</th>
<th>I</th>
<th>Info</th>
<th>W</th>
<th>Warning</th>
<th>E</th>
<th>Error</th>
<th>D</th>
<th>Debug</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I</td>
<td>Info</td>
<td>W</td>
<td>Warning</td>
<td>E</td>
<td>Error</td>
<td>D</td>
<td>Debug</td>
</tr>
</tbody>
</table>

The sample below shows a start of the new node

```
I 2018-12-31.16:48:58.838 Rules signature: 7e16d65b64ef2fbb
I 2018-12-31.16:48:58.986 starting a node on 10000 port...
I 2018-12-31.16:48:58.996 Node ID=5c8f92alcfaee337
I 2018-12-31.16:48:58.996 Initial Tip: 0-0000000000000000
I 2018-12-31.16:48:58.996 Requesting block 0-0000000000000000
I 2018-12-31.16:48:58.997 PI 0000000000000000--0.0.0.0 New
I 2018-12-31.16:48:58.997 PI 0000000000000000--0.0.0.0 Address changed to 23.239.24.209:8201
I 2018-12-31.16:48:58.999 stratum server listens to 0.0.0.0:10002
```

The node connects to the first peer, in this case 23.239.24.209:8201 and downloads the initial Tip at hight 0. It then requests the matching block. In this specific example the node also starts the Stratum server.

**CLI Wallet logs**

**Desktop Wallet logs**

**1.4.22 Consensus Rules**

The following parameters allow to configure consensus rules in Beam Node and Wallet

---

**WARNING: Used for development and testing only**

Consensus parameters are only relevant for testing and development purposes. Changing these parameters will change the Rule Signature and hence break the compatibility of the Node and Wallet with running Mainnet or Testnet servers.
## Rules

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description &amp; Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmissionValue0</td>
<td>Initial coinbase emission in a single block (in Groth, 10^-8 of Beam)</td>
</tr>
<tr>
<td></td>
<td><em>EmissionValue0=800000000</em></td>
</tr>
<tr>
<td>EmissionDrop0</td>
<td>Height of the last block that still has the initial emission.</td>
</tr>
<tr>
<td></td>
<td>Emission drops by half in the next block. Default equals 1 year, assuming 1 block per minute</td>
</tr>
<tr>
<td></td>
<td><em>EmissionDrop0=525600</em></td>
</tr>
<tr>
<td>EmissionDrop1</td>
<td>Number of blocks in halving cycle (defaults to four years, assuming 1 block per minute)</td>
</tr>
<tr>
<td></td>
<td><em>EmissionDrop1=2102400</em></td>
</tr>
<tr>
<td>MaturityCoinbase</td>
<td>Number of blocks that should be mined (confirmations) before coinbase UTXO can be spent.</td>
</tr>
<tr>
<td></td>
<td><em>MaturityCoinbase=240</em></td>
</tr>
<tr>
<td>MaturityStd</td>
<td>Number of blocks that should be mined (confirmations) before normal (non coinbase) UTXO can be spent.</td>
</tr>
<tr>
<td></td>
<td><em>MaturityStd=0</em></td>
</tr>
<tr>
<td>MaxBodySize</td>
<td>Block body size (in bytes)</td>
</tr>
<tr>
<td></td>
<td><em>MaxBodySize=0x100000</em></td>
</tr>
<tr>
<td>DesiredRate_s</td>
<td>Target block rate (in seconds)</td>
</tr>
<tr>
<td></td>
<td><em>DesiredRate_s=60</em></td>
</tr>
<tr>
<td>DifficultyReviewWindow</td>
<td>Number of blocks in the window for the mining difficulty adjustment</td>
</tr>
<tr>
<td></td>
<td><em>DifficultyReviewWindow=1440</em></td>
</tr>
<tr>
<td>TimestampAheadThreshold_s</td>
<td>Block timestamp tolerance (in seconds)</td>
</tr>
<tr>
<td></td>
<td><em>TimestampAheadThreshold_s=7200</em></td>
</tr>
<tr>
<td>WindowForMedian</td>
<td>Number of blocks considered in calculating the timestamp median</td>
</tr>
<tr>
<td></td>
<td><em>WindowForMedian=25</em></td>
</tr>
<tr>
<td>AllowPublicUtxos</td>
<td>Flag allowing regular (non-coinbase) UTXO to have non-confidential signature</td>
</tr>
<tr>
<td></td>
<td><em>AllowPublicUtxos=0</em></td>
</tr>
<tr>
<td>FakePoW</td>
<td>Flag to disable verification of PoW. Mining is simulated by timer.</td>
</tr>
<tr>
<td></td>
<td><em>FakePoW=0</em></td>
</tr>
</tbody>
</table>
Below is an example of corresponding .cfg file section:

```
# Rules configuration:

# initial coinbase emission in a single block
# EmissionValue0=800000000

# height of the last block that still has the initial emission, the drop is starting from the next block
# EmissionDrop0=525600

# Each such a cycle there's a new drop
# EmissionDrop1=2102400

# num of blocks before coinbase UTXO can be spent
# MaturityCoinbase=240

# num of blocks before non-coinbase UTXO can be spent
# MaturityStd=0

# Max block body size [bytes]
# MaxBodySize=0x100000

# Desired rate of generated blocks [seconds]
# DesiredRate_s=60

# num of blocks in the window for the mining difficulty adjustment
# DifficultyReviewWindow=1440

# Block timestamp tolerance [seconds]
# TimestampAheadThreshold_s=7200

# How many blocks are considered in calculating the timestamp median
# WindowForMedian=25

# set to allow regular (non-coinbase) UTXO to have non-confidential signature
# AllowPublicUtxos=0

# Don't verify PoW. Mining is simulated by the timer
# FakePoW=0
```

### 1.4.23 Local Setup

In some cases you would want to start a local network for testing and development purposes.

**Starting a new network**

To start a new Beam Network follow the steps below:

1. Download or build Beam binaries for node and CLI wallet

From this point the folders containing node and wallet binaries will be called ‘node folder’ and ‘wallet folder’ respectively

2. Initialize new wallet by running the following command in the wallet folder:
See **Creating new wallet** for more details

3. Export miner key, by running the following command in the wallet folder:

```
./beam-wallet export_miner_key --subkey=1
```

**Note:** If you are running more than one mining node repeat step 3 with –subkey=N (where N is the id of the mining nodes 1,2,3…)

4. Export owner key by running the following command in the wallet folder:

```
./beam-wallet export_owner_key
```

5. Download sample Beam treasury file `treasury.bin` (it is also located in the root of Beam source folder) and copy it to the same folder as beam-node binary.

6. Launch the first node using the following command:

```
./beam-node --treasury_path=treasury.bin --owner_key=<owner key> --miner_key=<miner key> --pass=<wallet password>
```

Sample output printed by the node to the console (and in the logs) should look like this:

```
I 2018-12-25 09:43:56.712 Rules signature: ddccf5d8d0f77bd2
I 2018-12-25 09:43:56.925 starting a node on 10000 port...
I 2018-12-25 09:43:56.930 Treasury size: 584661
I 2018-12-25 09:43:56.955 Node ID=a82306ff757cca58
I 2018-12-25 09:43:56.955 Initial Tip: 0-000000000000000000000000000000
I 2018-12-25 09:43:56.955 Sync mode
I 2018-12-25 09:43:56.955 Searching for the best peer...
I 2018-12-25 09:43:56.959 GenerateNewBlock: size of block = 294; amount of tx = 0
I 2018-12-25 09:43:56.960 Block generated: Height=1, Fee=0, Difficulty=02-000000(4), Size=294
I 2018-12-25 09:43:56.960 Mining nonce = 428dfc03b88fa57d
```

As new blocks are mined, the console will indicate this as follows:

```
I 2018-12-25 09:46:00.513 New block mined: 1-a690c8aa7e14225c
I 2018-12-25 09:46:00.520 1-a690c8aa7e14225c Header accepted
I 2018-12-25 09:46:00.520 1-a690c8aa7e14225c Block received
I 2018-12-25 09:46:00.523 1-a690c8aa7e14225c Block interpreted. Fwd=1
I 2018-12-25 09:46:00.524 My Tip: 1-a690c8aa7e14225c, Work = 4
I 2018-12-25 09:46:00.527 GenerateNewBlock: size of block = 294; amount of tx = 0
I 2018-12-25 09:46:00.527 Block generated: Height=2, Fee=0, Difficulty=02-000000(4), Size=294
I 2018-12-25 09:46:00.528 Mining nonce = 32d6ce8dba24d0dd
```

**Note:** For GPU mining, add –miner_type=gpu parameter

7. Launch additional nodes

To launch more mining nodes either for the same wallet or different ones, just repeat the relevant portions of steps 1 through 6.
In addition add –peer=<ip:port of the first node> to connect new nodes to the first node, thus extending the network

**Attention:** Beam Nodes have automatic discovery mechanism that will find and automatically connect to the other running nodes on the local network. If you do not want the networks to mix, just change any of the *Consensus Rules* to intentionally break compatibility between the networks and avoid unwanted behavior. The simplest way to achieve this is to set ‘Prehistoric’ parameter to a random hash value, i.e Prehistoric=1234