Eindhoven University of Technology
Department of Mathematics and Computer Science

2WH40 Bachelor Final Project

User manual for Brabocoin

An educational cryptocurrency based on Bitcoin

April 9, 2019
Version 0.4

Authors

D.P. van den Berg	0949036
D.J.C. Dekker	0936100
S. van den Eerenbeemt	0954445
S. Wessel	0941508

Supervisors
B.M.M. de Weger
B. Skoric

Contents

1 Overview

Bitcoin is developed to be a payment system that allows online payments between parties without a central authority: a decentralized peer-to-peer cash system or *cryptocurrency*. Because the design of Bitcoin is based on cryptographic proofs instead of trust, Bitcoin is based on various mathematical principles. Bitcoin is designed to be comprehensible by the average user, which means the cryptography is hidden in the internal functionality of the system. Its software is an open source project, meaning its design is public. The Bitcoin core has matured and experienced multiple adjustments in order to add features and improve efficiency, making it challenging to understand the principles and foundation it is based on.

This suggests the need for an educational implementation of a cryptocurrency based on Bitcoin, which exposes both the inner workings and the mathematics that supports the Bitcoin system. Such a tool provides a way for interested parties, such as Bitcoin enthousiasts and students, to explore and experiment with the structure and functioning of Bitcoin.

To this end, the Brabocoin application was developed. It is based on the foundation of Bitcoin and provides interested parties with an environment where they can explore and experiment with the workings of cryptocurrencies. In order to find out more about the Brabocoin application and its differences compared to Bitcoin, see the report available at `https://brabocoin.org`.

This manual provides an overview of all functionality present in the Brabocoin application. In Section 2, every page of the Brabocoin application is described. All pages are displayed with a short description and a list of all functionality available on the page. In Section 3, a list of short tutorials is provided. Each tutorial explains in detail how to use one specific aspect of the application.

We do not recommend reading the entire manual. Instead, the manual should be used as a point of reference, in case something in the application is unclear.

2 Page descriptions

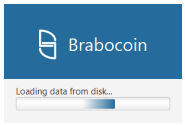
This section describes each page of the Brabocoin application in detail. Section 2.1 provides descriptions of the pages belonging to initialization, startup and the general overview of the Brabocoin application. The remaining sections each cover one of Brabocoin's main menu items, including all pages that belong to this menu item.

2.1 General

In this section, the initialization and startup pages of the Brabocoin application are covered. Afterwards, a general overview of the Brabocoin application is provided, together with the application log and the settings menu.

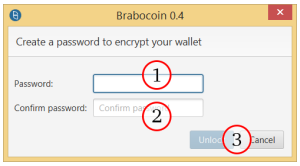
2.1.1 Starting the application

When the Brabocoin application is started, the application will first load data from disk. This is indicated by the progress bar displayed on the loading screen. When loaded, either a *Password creation* dialog or an *Unlock wallet* dialog will be displayed (see Sections 2.1.2 and 2.1.3), depending on whether an existing wallet was already saved on disk.



2.1.2 Password creation

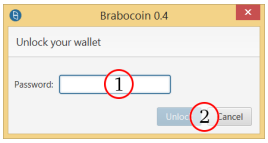
When the Brabocoin application is started for the first time, a new wallet must be created. To this end, a password must be entered, which is used to secure the wallet. This password must be filled in every time the Brabocoin application started.



1. In the *Password* field, a password must be entered.
2. In the *Confirm password* field, the same password must be entered again.
3. The *Unlock* button creates a wallet with the provided password. The *Cancel* button closes the application.

2.1.3 Unlock wallet

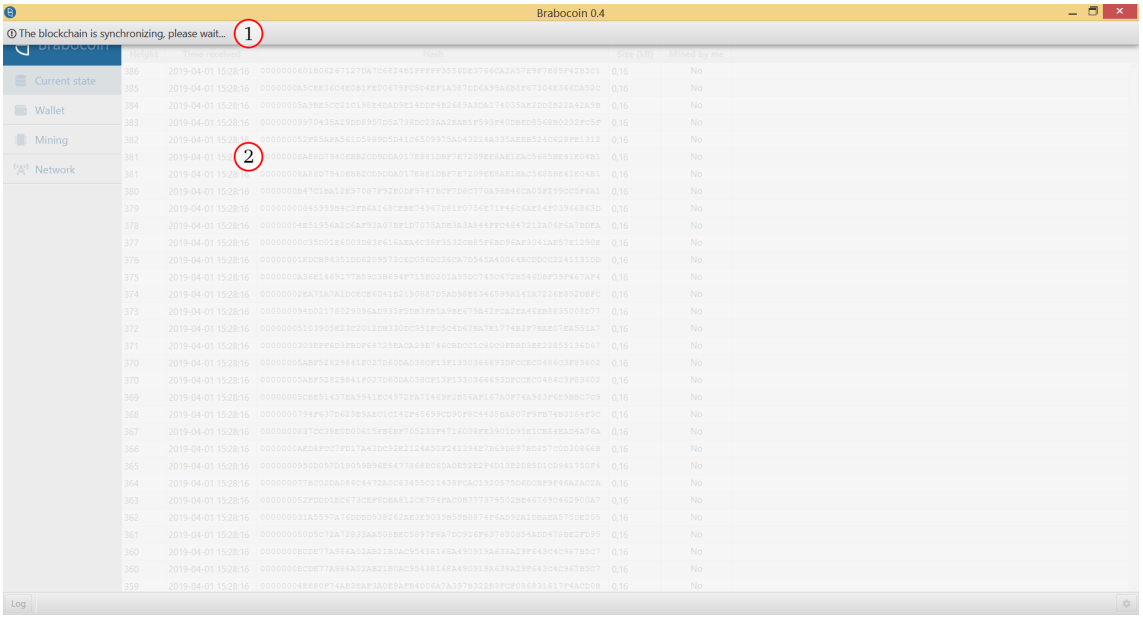
When the Brabocoin application is started with an existing wallet, the wallet must be unlocked. To this end, the password used to secure the wallet must be entered.



1. In the *Password* field, the password used to secure the wallet, must be entered.
2. The *Unlock* button attempts to unlock the wallet with the provided password. The *Cancel* button closes the application.

2.1.4 Synchronizing

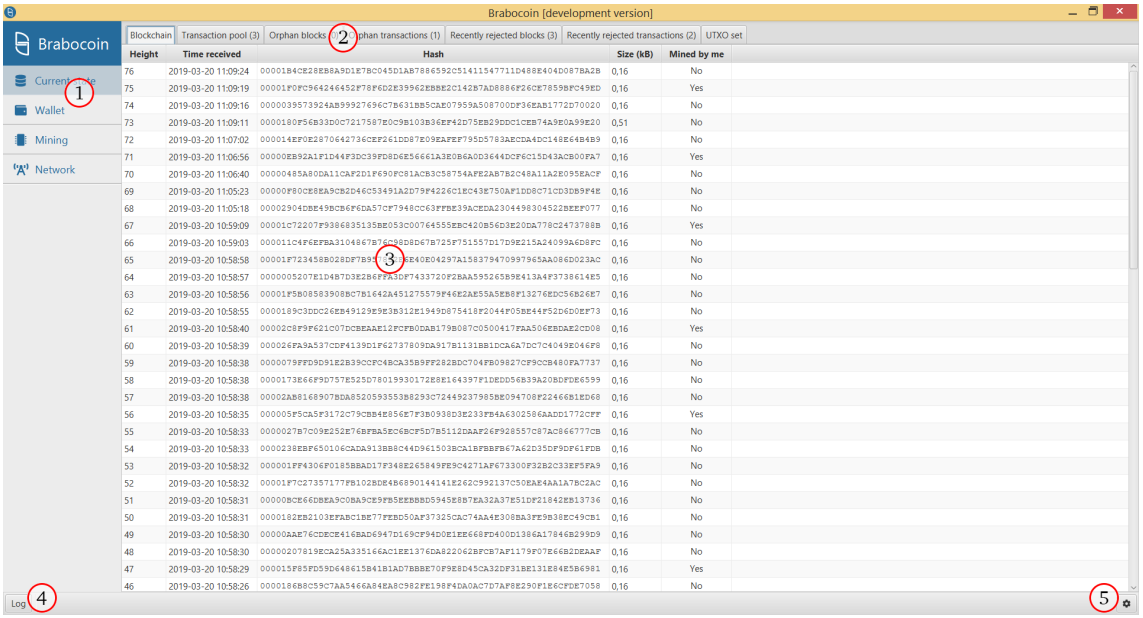
Once the wallet is unlocked (see Section 2.1.3), the application will automatically start synchronizing the blockchain. During the synchronization process, the entire application is blocked and cannot be used yet.



1. This message indicates that your blockchain is synchronizing.
2. The entire application is grey and can not be used yet.

2.1.5 General overview

When the Brabocoin application is opened and the blockchain is synchronized (see Section 2.1.4), a page is displayed showing the current state of the blockchain. A main overview of the Brabocoin application is now provided.



1. On the left, the Brabocoin logo and navigation menu are displayed. The navigation menu contains the following menu items:
    - *Current state* (see Section 3.4) provides information on the current state of your Brabocoin application. Among others, the transaction pool and blockchain can be viewed here. Next

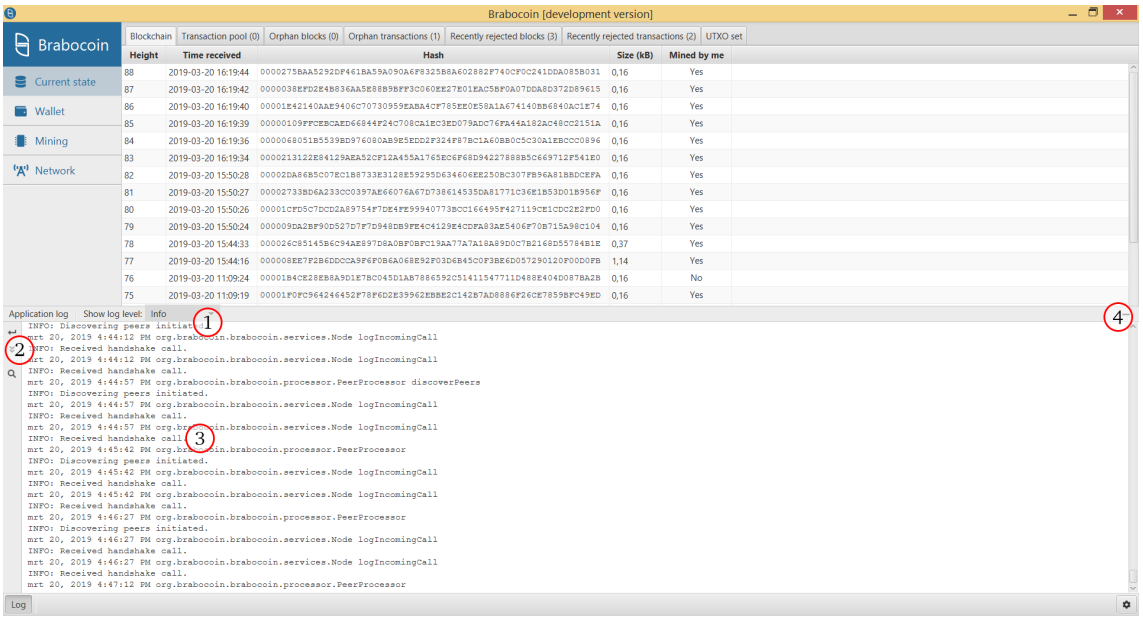
to this, rejected blocks and transactions can be viewed and blocks and transactions can be validated.

- *Wallet* (see Section 2.3) displays a list of key pairs and information on your balance and your transaction history. Here, new transactions can be created.
- In *Mining* (see Section 2.4), new blocks can be mined.
- *Network* (see Section 2.5) displays your network information. All messages that are sent to and received from your peers can be viewed. Peers can be added or removed.

2. When a menu item is selected, the main view will switch to that menu item. The functionality for this menu item is divided into multiple tabs. These tabs differ for each menu item and are described in the section corresponding to the menu item.
3. The content of the chosen tab is displayed in the middle of the page. In the example figure, the blockchain page is displayed.
4. In the bottom-left corner, you can click the *Log* button to open or close the application log. This log keeps track of everything that the application is doing. See Section 2.1.6 for more information.
5. In the bottom-right corner, you can click the *Settings* icon to open the settings dialog. Here, you can change network independent configuration values like your port number, but also network dependent consensus values such as the target value. See Section 2.1.7 for more information.

2.1.6 Logging

The application log describes everything that the Brabocoin application is doing, up to a chosen level of detail.

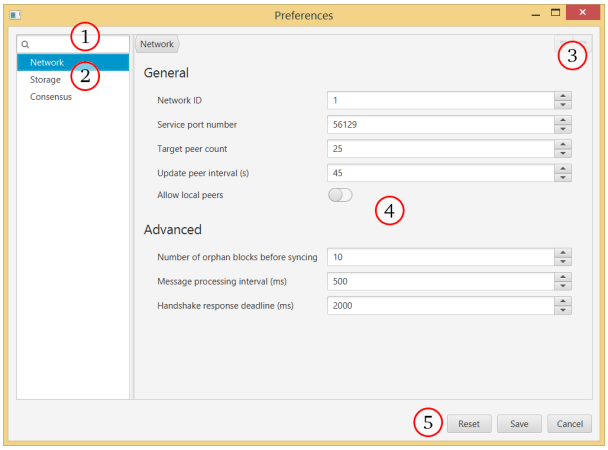


1. In the *Show log level* dropdown, a log level can be selected. Only log statements that have an equal or higher level are recorded in the log. For example, if *Info* is chosen as level, only *Info*, *Warning* and *Severe* log statements are displayed.
2. The following tools can be used to read the logs:
   - The arrow icon toggles line wrapping. If line wrapping is enabled, lines that do not fit on the screen are split onto the next line.
   - The double arrowhead icon scrolls down to the most recent log statement.
   - With the search icon, it is possible to search through the logs using a specific keyword. Regular expressions can also be used to search.

3. Each log statement starts with the date and time it was recorded, followed by the Java class and method that placed the log statement. On the next line, the log level and description of the log statement are displayed.
4. By clicking the minimization icon in the top-right corner, the application log is closed.

2.1.7 Settings

In the *Settings* dialog, some settings regarding the network, storage and the consensus of Brabocoin can be changed. Beware that changing consensus values might invalidate incoming blocks and transactions on your current network (see Section 2.2.5). An easy recovery procedure is to uninstall the Brabocoin application to get rid of the corrupted data.



1. In the search bar, you can search through the different settings to find a specific value.
2. Below the search bar, the following three groups of settings are displayed:
   - *Network* settings, such as the port and network ID.
   - *Storage* settings, such as folder names and size limits.
   - *Consensus* settings, such as the target value. Changing consensus values allows the creation of invalid blocks and transactions. Be aware that this might corrupt your data in your current network (see Section 2.2.5).
3. Using the *Undo* and *Redo* buttons, the last changes can be undone and redone.
4. In the middle, the settings belonging to the chosen group are displayed. Each setting displayed here can be changed.
5. The *Reset* button resets all settings back to their original value. The *Save* button saves the changes made, which requires restarting the application for the changes to be applied. The *Cancel* button discards the changes made and closes the *Settings* dialog.

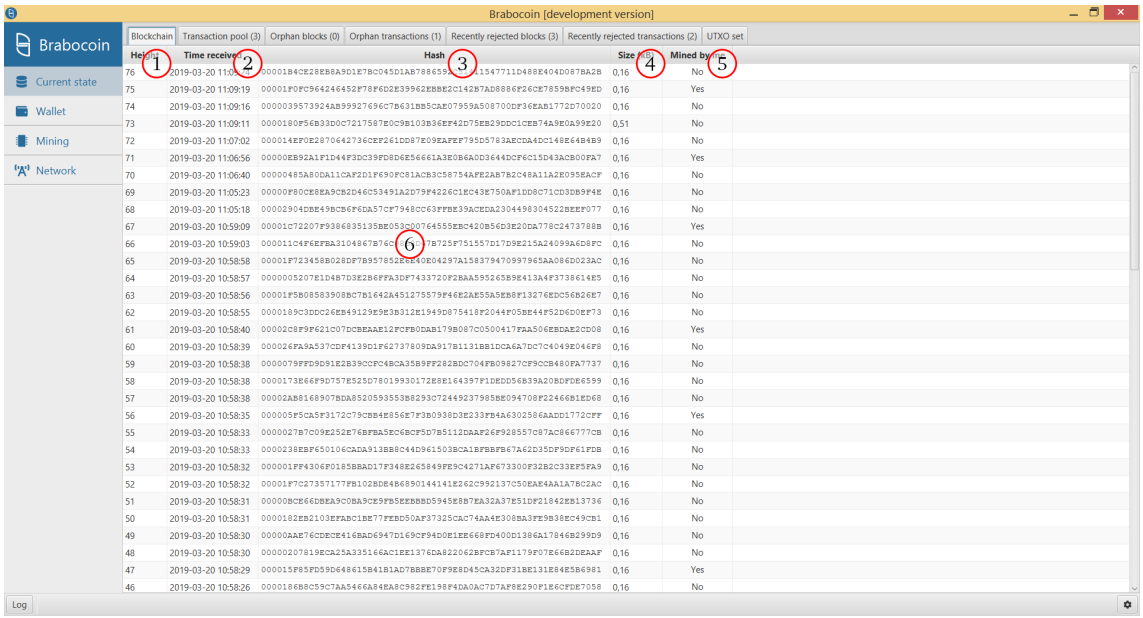
2.2 Current state

By clicking *Current state* in the navigation menu, as explained in Section 2.1.5, information is displayed on the current state of your Brabocoin application. The following tabs can be opened:

- *Blockchain*: this tab opens the blockchain page. See Section 2.2.1 for more information.
- *Transaction pool*: this opens the transaction pool, which contains all valid transactions that are not yet included in a block in the blockchain. The trailing number in parentheses indicates the amount of transactions currently in the transaction pool. See Section 2.2.2 for more information.
- *Orphan blocks*: this page displays all orphan blocks, which are blocks that reference an unknown parent block (also called the *previous block*). The trailing number in parentheses indicates the current amount of orphan blocks. See Section 2.2.3 for more information.
- *Orphan transactions*: this page displays orphan transactions, which are transactions that reference an unknown input. The trailing number in parentheses indicates the current amount of orphan transactions. See Section 2.2.4 for more information.
- *Recently rejected blocks*: this page displays blocks that are invalid. The trailing number in parentheses indicates the current amount of recently rejected blocks. See Section 2.2.5 for more information.
- *Recently rejected transactions*: this page displays transactions that are invalid. The trailing number in parentheses indicates the current amount of recently rejected transactions. See Section 2.2.6 for more information.
- *UTXO set*: this page displays the set of all transaction outputs that are currently unspent. See Section 2.2.7 for more information.

2.2.1 Blockchain page

The *Blockchain* page will be displayed after the application was started and the wallet was unlocked (see Section2.1.3). It displays the main chain of the current network, which is the longest chain of blocks in the current blockchain. Notice that not the entire blockchain can be viewed: forks are not displayed.



1. The *Height* column displays the height of the block in the main chain.
2. The *Time received* column displays the time at which the block was received. Notice that this is not always equal to the time at which the block was mined. New blocks could have been mined while the node was offline. These blocks are received at the moment you start the application and the blockchain is synchronized.

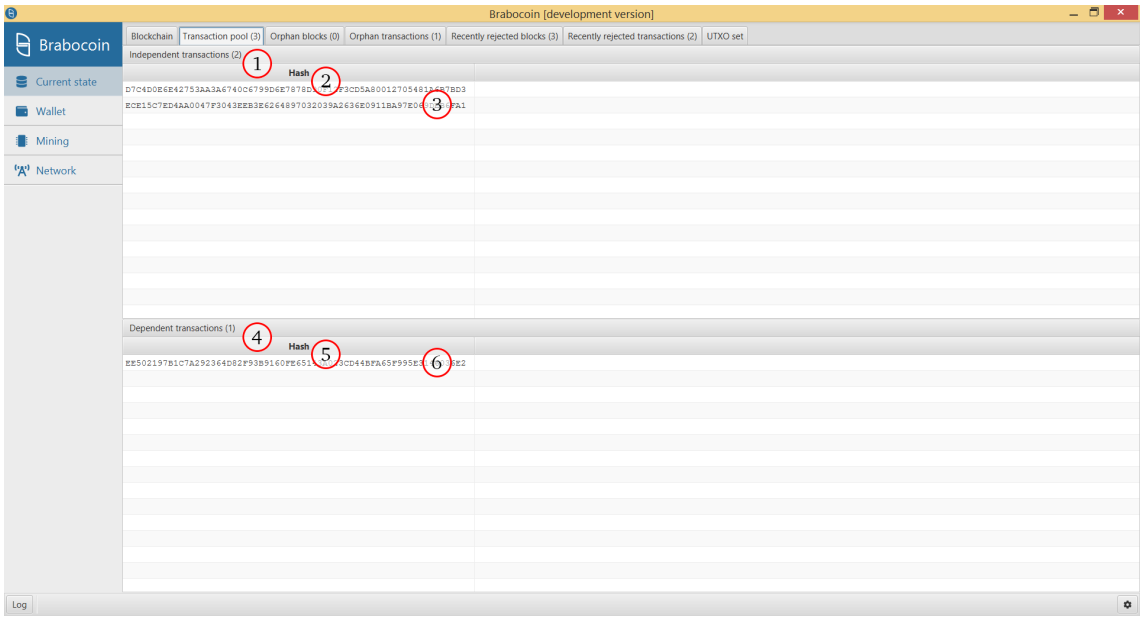
3. The *Hash* column displays the hash of the block. This is equal to the double SHA-256 hash of the data in the block header of the block. The block header of a block can be viewed on the page explained in Section 2.2.8. Notice that each valid block hash starts with a number of zeros, because the proof-of-work algorithm states that the block hash of a valid block must be smaller than a specified target value.
4. The *Size* column displays the size of the block in kilobytes.
5. The *Mined by me* column displays whether the block was mined by you, or not.
6. When clicking on block in the table, the *Block detail view* side pane (see Section 2.2.8) will open on the right. This pane shows detailed information on the block that was clicked. If the row is clicked again, the *Block detail view* side pane will close.

2.2.2 Transaction pool

When a transaction is created, it is placed in the transaction pool before someone mines a block including the created transaction. All transactions in the transaction pool are valid, but they are not yet included in the main chain of the blockchain.

The transaction pool is divided in two parts: *independent transactions* and *dependent transactions*. Independent transactions can be mined immediately: they only reference transactions that are already included in a block in the blockchain. Dependent transactions, on the other hand, reference one or more transactions that are still in the transaction pool. These dependent transactions can only be mined once the referenced transactions in the transaction pool have been mined.

Note that when the application is closed, all transactions in the transaction pool are discarded. When the Brabocoin application is started, the transaction pool is synchronized with other nodes in the network.



1. In the *Independent transactions* section, all independent transactions in the pool are displayed. In the description above is explained which transactions are independent. The number between parentheses indicates the amount of independent transactions in the transaction pool.
2. The *Hash* column displays the hash of the independent transaction. This is equal to the double SHA-256 hash of the data of the (signed) transaction.
3. Each row represents an independent transaction in the transaction pool. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This pane shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.
4. In the *Dependent transactions* section, all dependent transactions in the transaction pool are displayed. In the description above is explained which transactions are considered dependent. The

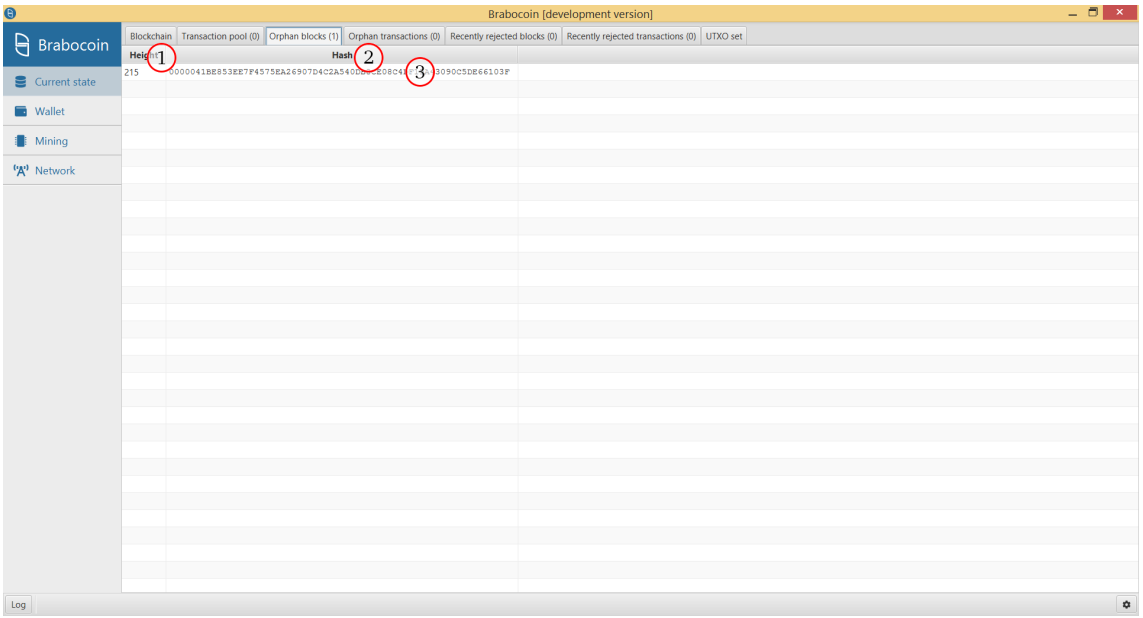
number between parentheses indicates the amount of dependent transactions in the transaction pool.

5. The *Hash* column displays the hash of the dependent transaction. This is equal to the double SHA-256 hash of the data of the (signed) transaction.

6. Each row represents a dependent transaction in the transaction pool. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.

2.2.3 Orphan blocks

Each block in the blockchain contains a reference to a parent block in the blockchain. This parent block is referenced by including the previous block hash in the block header of the current block (see Section 2.2.8). Note that it is this reference that constructs a chain of blocks. It could, however, happen that a block is received that contains a reference to a parent block that is unknown. For example, the parent block might have been missed, or might not even exist at all. Such a block, with an unknown previous block hash, is called an orphan block, and these blocks are displayed on the *Orphan blocks* page. Note that when the application is closed, all orphan blocks are discarded.



1. The *Height* column displays the height of the orphan block, which is stored in the block header of the orphan block.

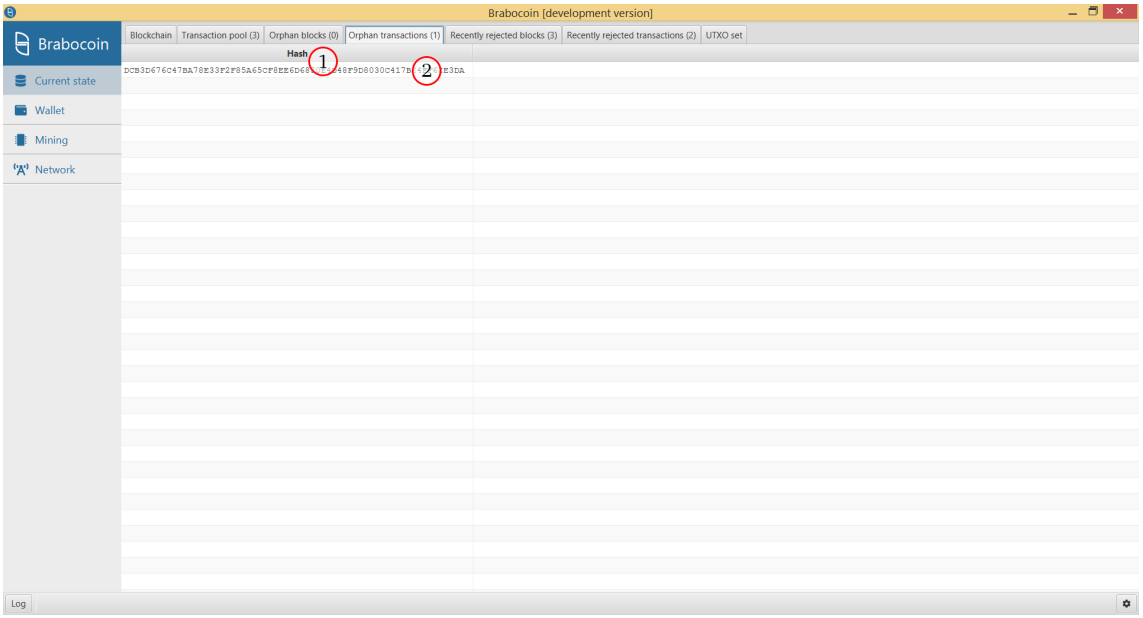
2. The *Hash* column displays the hash of the block. This is equal to the double SHA-256 hash of the data in the block header.

3. Each row represents an orphan block. When such a row is clicked, the *Block detail view* side pane (see Section 2.2.8) will open on the right. This shows detailed information on the block that was clicked. If the row is clicked again, the *Block detail view* side pane will close.

2.2.4 Orphan transactions

Each transaction contains one or more references to outputs of earlier transactions. These form the inputs that are being spent in this transaction. This reference includes the hash of the (signed) transaction, and the index of the referenced output in the transaction. It could, however, happen that a transaction contains a reference to some transaction output that is unknown. For example, the referenced transaction might have been missed, or might not even exist at all. Such a transaction with an unknown reference is called an orphan transaction, and these transaction are displayed on the *Orphan transactions* page. Notice that indeed an orphan transaction could also be invalid if the referenced transaction does not exist at all. However, since at the time we cannot distinguish between an invalid

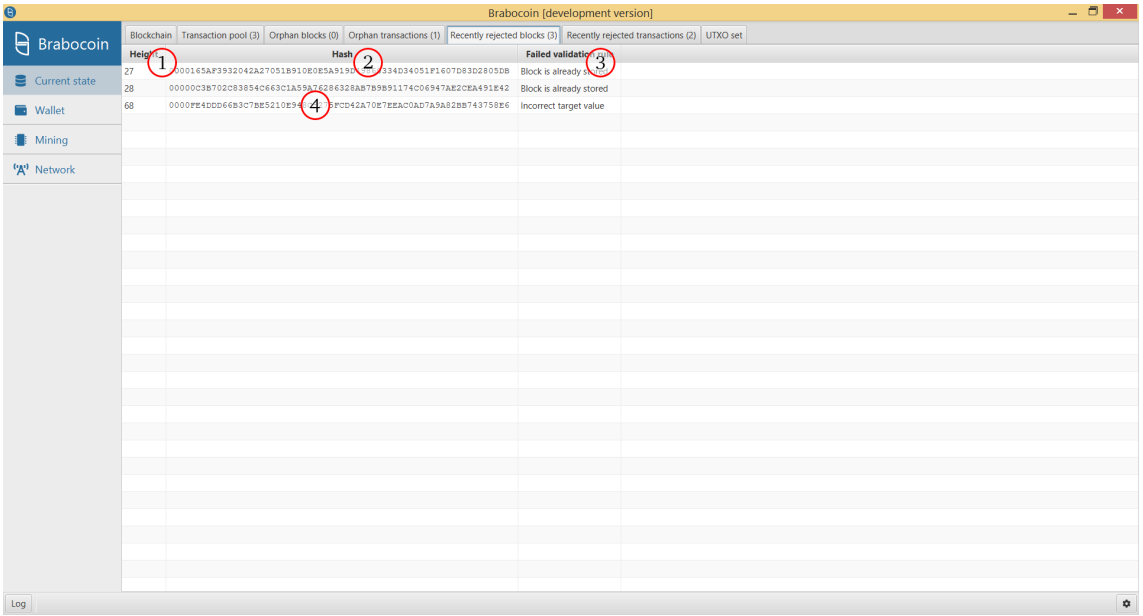
orphan transactions or a missed orphan transaction, no orphan transaction is seen as invalid. When the application is closed, all orphan transactions are discarded.



1. The *Hash* column displays the hash of the orphan transaction. This is equal to the double SHA-256 hash of the data of the signed transaction.
2. Each row represents an orphan transaction in the transaction pool. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.

2.2.5 Recently rejected blocks

In the Brabocoin application, it is possible to create invalid blocks. For example, the target value can be changed (see Section 2.1.7) to make mining easier. Other users will reject the block, because it has an incorrect target value and is therefore invalid. These rejected blocks are displayed on the *Recently rejected blocks* page. Note that when the application is closed, all rejected blocks are discarded.



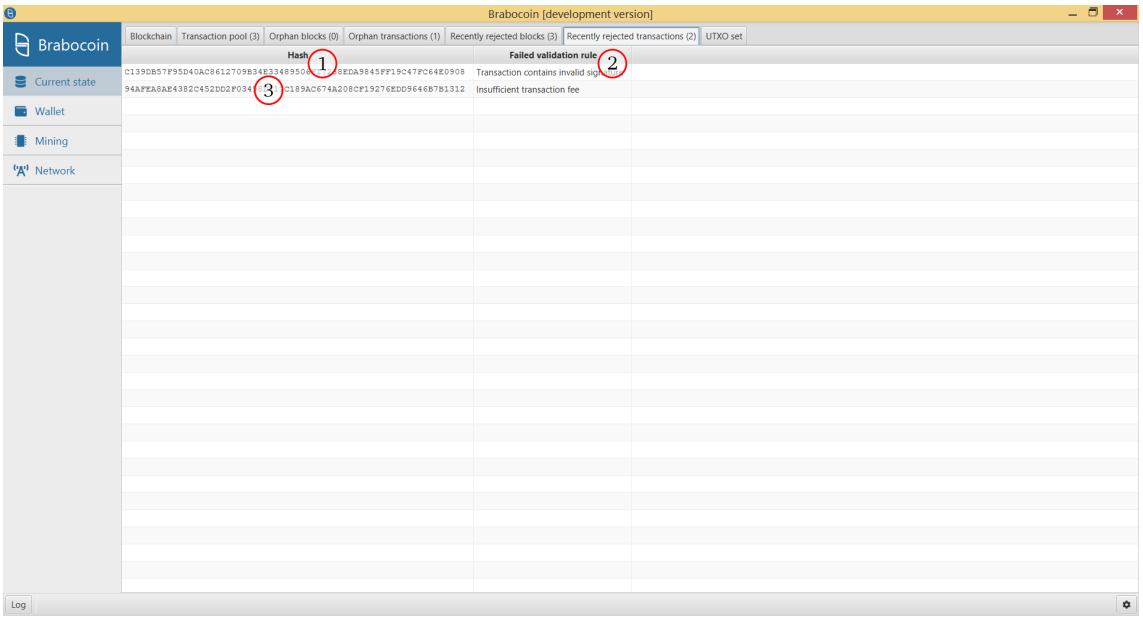
1. The *Height* column displays the height of the rejected block in the main chain.
2. This is equal to the double SHA-256 hash of the data in the block header.

3. The *Failed validation rule* column indicates why the block is invalid. In the image above, two blocks were invalid because they were already stored. This happens if you receive a certain block from one peer first and later from another peer. The third block in the image above is invalid because was mined with an incorrect target value. If you want more information on the validation of a block, you can validate the block again by clicking the *Validate* button (see Sections 3.4.4).

4. Each row represents a rejected block. When such a row is clicked, the *Block detail view* side pane (see Section 2.2.8) will open on the right. This shows detailed information on the block that was clicked. If the row is clicked again, the *Block detail view* side pane will close.

2.2.6 Recently rejected transactions

In the Brabocoin application, it is possible to create invalid transactions. For example, you could spend more money in the transaction outputs than you referenced in the transaction inputs. Other users will reject the transaction, because it is invalid. These rejected transactions are displayed on the *Recently rejected transactions* page. Note that when the application is closed, all rejected transactions are discarded.



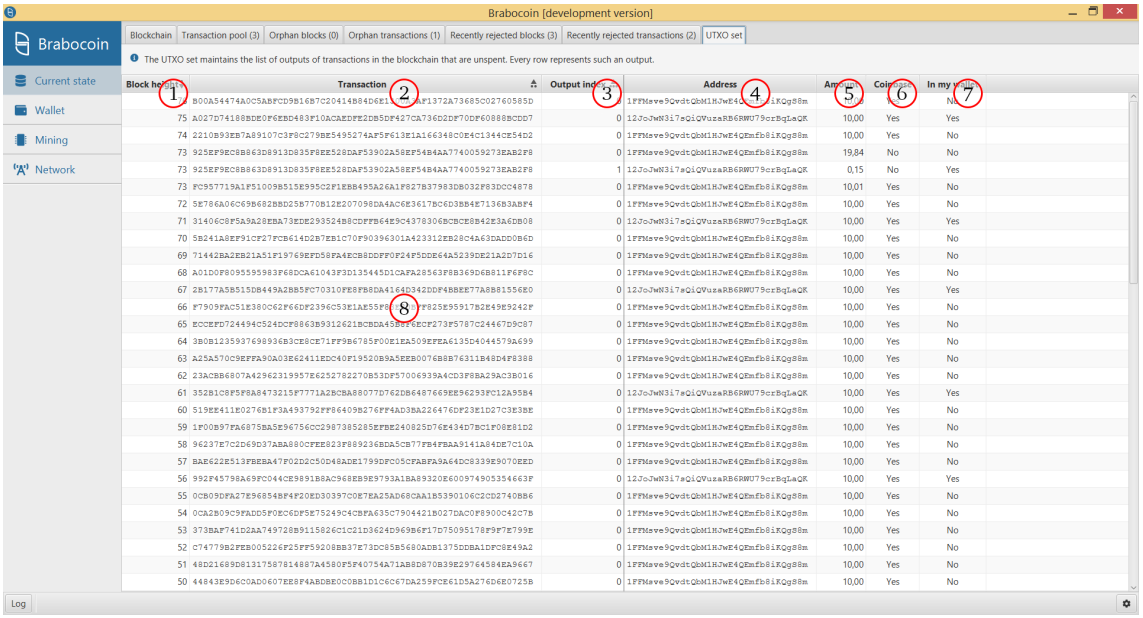
1. The *Hash* column displays the hash of the rejected transaction. This is equal to the double SHA-256 hash of the data of the (signed) transaction.

2. The *Failed validation rule* column indicates why the transaction is invalid. In the image above, one transaction is invalid, because it contains invalid signatures. If you want more information on the validation of transactions, you can validate a transaction again by clicking the *Validate* button (see Section 3.4.5).

3. Each row represents a rejected transaction. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.

2.2.7 UTXO set

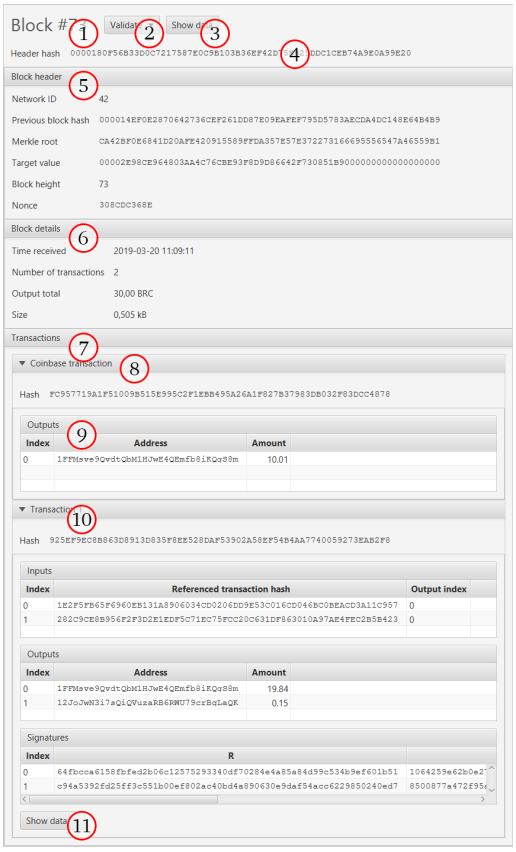
All inputs of a transaction must reference a transaction output that is not yet spent. It does not only take much time to find the referenced transaction output in the blockchain, but it is also expensive to check that the referenced output is not already spent in another transaction. To speed up this process, the application keeps track of a so-called *UTXO set*: an index of all unspent transaction outputs. The outputs in the UTXO set are displayed on the *UTXO set* page. Note that on this page, it is possible to sort the UTXO set on a property by clicking on the name of the column you wish to sort on.



1. The *Height* column displays the height of the block that contains the unspent transaction output.
2. The *Hash* column displays the hash of the transaction that contains the unspent transaction output. This is the double SHA-256 hash of the data of the signed transaction.
3. The *Output index* column displays the index of the unspent transaction output in the transaction it is contained in.
4. The *Address* column displays the address to which the unspent transaction output transferred. Note that only the owner of this address can spend the unspent transaction output. The address is displayed in `Base58Check` format.
5. The *Amount* column displays the amount of brabocoin that is transferred in this unspent transaction output.
6. The *Coinbase* column indicates whether the unspent transaction output is from a coinbase transaction. The coinbase transaction of a block pays the mining reward and transaction fees to the miner of the block.
7. The *In my wallet* column indicates whether the transaction output was paid to an address of which your wallet contains the corresponding private key. If this column displays *yes*, this means you can spend this unspent transaction output.
8. Each row represents an unspent transaction output. All information on this output is provided in the table.

2.2.8 Block detail view

The *Block detail view* displays all information that is stored in a block. Next to this, blocks can be validated in this view and the raw data of the block can be displayed. This raw data can be used to calculate, for example, the block hash of the block.



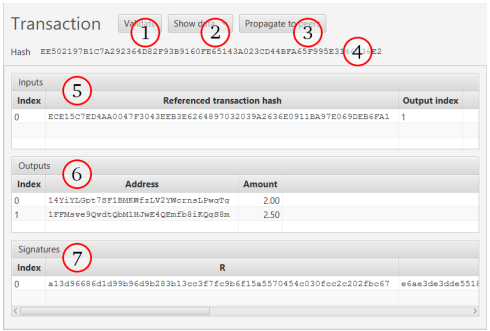
1. *Block #73* indicates the block height of the block, which is 73 in this example.
2. The *Validate* button allows you to validate the displayed block. After clicking this button, you can choose between *Quick* and *Complete*. The quick validation skips some validation checks, because these checks are expensive to verify once a block is processed and included in the blockchain. The complete validation performs all validation checks. Note that when the blockchain is long, complete validation may take a while to finish. Both options will open a *Validation* dialog (see Section 2.2.11).
3. The *Show data* button opens a *Data view* dialog (see Section 2.2.10). This dialog displays the raw data of the block, which can be used to calculate the block hash of the block.
4. The *Header hash* is the double SHA-256 hash of the information in the block header and is often called the *Block hash*.
5. Under *Block header*, all information that is stored in a block header is displayed:
   - The *Network ID* displays the network ID of the block. Only blocks with the same network ID as defined in your settings are processed; other blocks are seen as invalid on this network.
   - The *Previous block hash* is a reference to a previous block in the blockchain. It is this reference that constructs a chain of blocks.
   - The *Merkle root* is the root of the Merkle tree of all transaction hashes contained in the block.
   - The *Target value* displays the target value that was used to mine the block. This indicates the difficulty of the proof-of-work algorithm used to mine the block. Notice that, for a block to be valid, the header hash has to be smaller than the set target value.
   - The *Block height* is displayed again.
   - The *Nonce* is an arbitrary integer, represented in hexadecimal format. The nonce is varied in the proof-of-work algorithm, in order to find a block hash smaller than the set target value. The nonce thus allows randomness in blocks, which makes mining possible.
6. Under *Block details*, some additional information on the block is displayed. This information is not stored in the block itself:

- The *Time received* displays the date and time at which the block was received. Notice that this is not always equal to the time at which the block was mined. If the node was offline for some time, blocks will be received at a later time.
- The *Number of transactions* counts the transactions in the block. Note that this includes the coinbase transaction of the block.
- The *Output total* provides the total sum of the output amounts of all transactions in the block.
- The *Size* displays the size of the block in kilobytes.

7. Under *Transactions*, all transactions in the block are displayed. The block in the example above contains two transactions: one coinbase transaction and one regular transaction.
8. Under *Coinbase transaction*, the coinbase transaction of the block is displayed. First, the hash of the transaction is displayed. It only contains an output that transfers the mining reward and transaction fees to the miner of the block. The information can be collapsed and expanded by clicking the displayed triangle.
9. The coinbase output contains an index (which is always 0), an address and an amount. The address is the address where the mining reward and transactions fees are paid to. The amount consists of the sum of the mining reward (10 BRC in the example above) and the transaction fees (0.01 BRC in the example above).
10. Under *Transaction 1*, the *Transaction detail view* of the first non-coinbase transaction is displayed. See Section 2.2.9 for more information. Note that in other blocks, more transactions could be displayed underneath, if the block contains more regular transactions.
11. With the *Show data* button, it is possible to see the raw data of the displayed transaction. This raw data can be used to manually calculate the hash of the transaction or the signatures for the transaction.

2.2.9 Transaction detail view

The *Transaction detail view* displays all information that is stored in a transaction. Next to this, transactions can be validated in this view and the raw data of the transaction can be displayed. This raw data can be used to, for example, calculate the transaction hash of the transaction or verify the transaction's signatures.

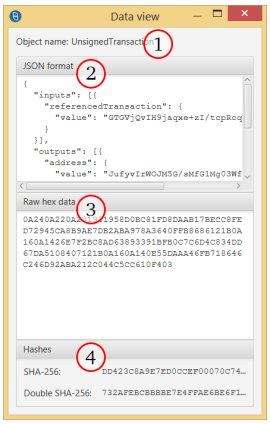


1. The *Validate* button allows you to validate the displayed transaction. This will open the *Validation* dialog (see Section 2.2.11).
2. The *Show data* button opens the *Data view* dialog (see Section 2.2.10). This dialog displays the raw data of the transaction. This raw data can be used to manually calculate the transaction hash or verify signatures.
3. The *Propagate to peers* button will send the transaction to all your peers. In case you receive a valid transaction, this will happen automatically. However, invalid transactions are not automatically propagated, and can thus be propagated manually using this button.
4. The *Hash* is the hash of the displayed transaction. This is the double SHA-256 hash of the data of the signed transaction.

5. Under *Inputs*, all inputs, i.e. referenced transaction outputs, are displayed. Each input contains the index of the input, the transaction hash of the referenced transaction and the output index of the referenced unspent transaction output.
6. Under *Outputs*, all outputs of the transaction are displayed. Each output contains the output index, the address of the recipient and the amount of brabocoin that is transferred.
7. Under *Signatures*, all signatures of the transaction are displayed. Each input must be signed using a digital signature, which proves that you may spend the referenced unspent transaction output. Each signature has an index and consists of three values, which together form the digital signature: R, S and a public key.

2.2.10 Data view

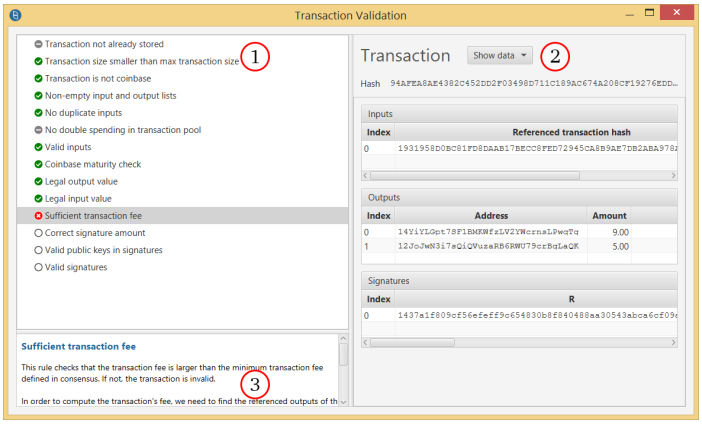
The *Data view* dialog displays the raw data of blocks and transactions. This raw data can be used to manually verify hashes, or calculate signatures.



1. The *Object name* gives the type of object that is opened. This can be a block, a signed or unsigned transaction, or a message (see Section 2.5.2). Note that the raw signed transaction data can be used to calculate a transaction hash, while the raw unsigned transaction data can be used to calculate the signatures of a transaction.
2. Under *JSON format*, the data is provided in a human-readable format.
3. Under *Raw hex data*, a sequence of hexadecimal integers is displayed. This is the raw data of the object, which can be used to calculate a hash or a signature manually.
4. Under *Hashes*, the SHA-256 hash and the double SHA-256 hash of the raw hex data are displayed.

2.2.11 Validation

To get an understanding of the validation of transactions and blocks, the *Validation dialog* can be used. This dialog displays all rules that are checked in the validation process. Each rule also includes a descriptions.



1. In the top-left corner of the dialog, all validation rules are displayed. The icon in front of the rule indicates the result of this validation step. A green checkmark icon indicates a passed rule and a red cross icon indicates a failed rule. Sometimes it is impossible to check a validation rule at this time, in which case the rule is skipped. Skipped rules are indicated with a grey horizontal bar icon. Once one rule has failed, remaining rules are not checked which is indicated with a white icon.
2. On the right hand side of the dialog, the *Transaction detail view* of the transaction that is being validated, is displayed (see Section 2.2.9).
3. In the bottom-left corner of the dialog, a description of the selected rule is displayed.

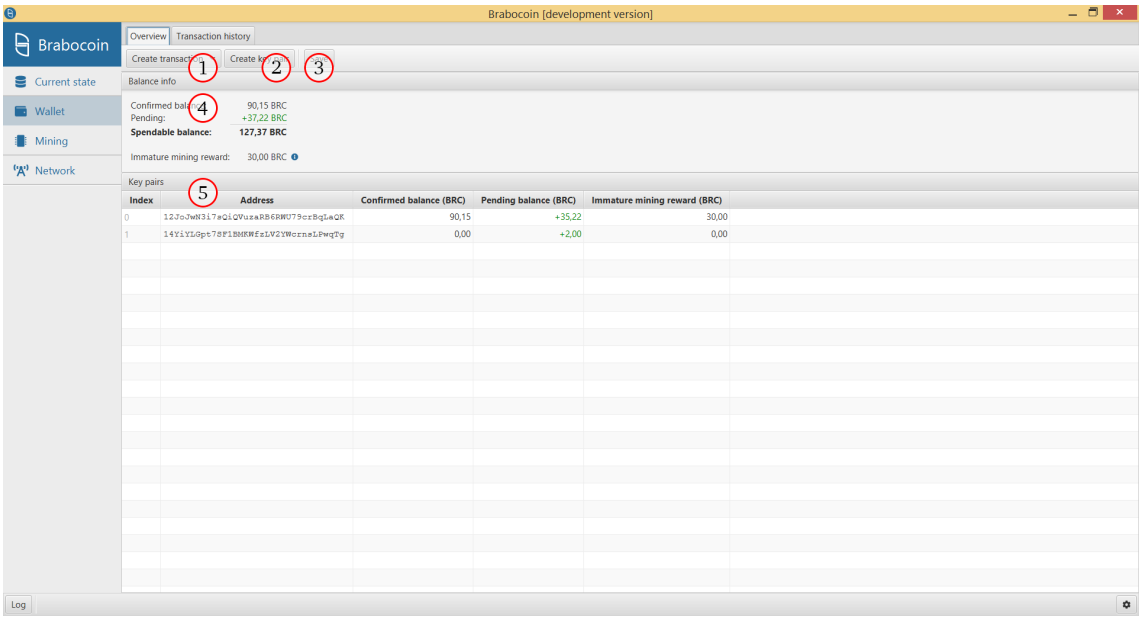
2.3 Wallet

By clicking the *Wallets* menu item in the navigation menu (see Section 2.1.5), your wallet is displayed. The *Wallets* menu item contains the following two tabs:

- The *Overview* tab provides information on your balance and keys. Transactions can be created in this tab.
- The *Transaction history* tab contains a list of all transactions that involve keys from your wallet.

2.3.1 Wallet overview

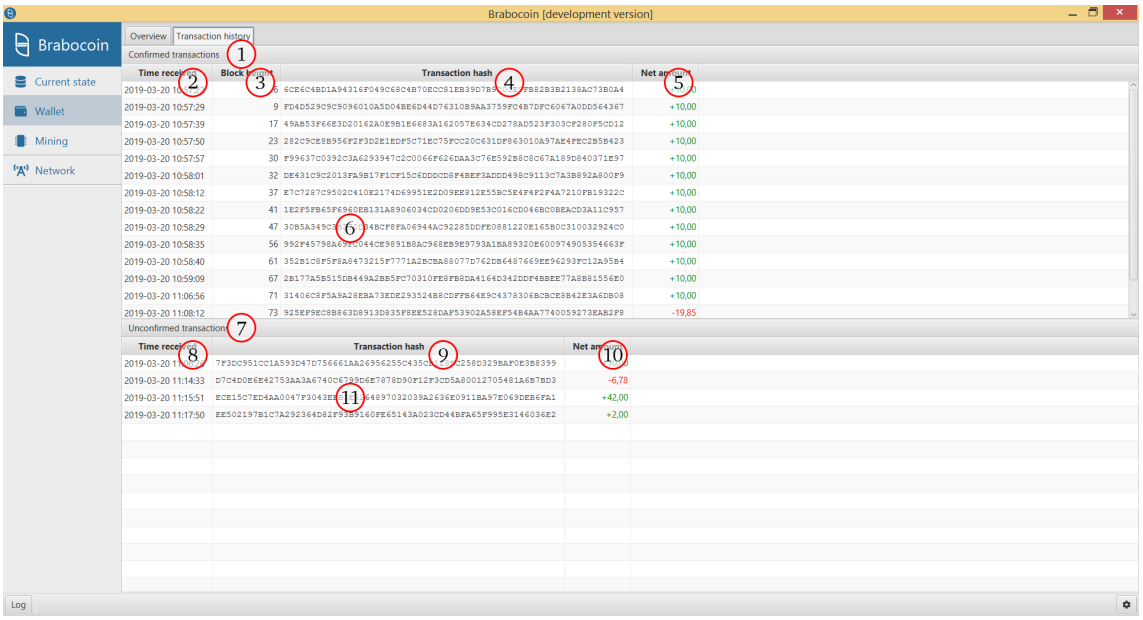
In the *Wallet overview* tab, information on your balance and key pairs is displayed.



1. With the *Create transaction* button, a new transaction can be created. After clicking the button, you can choose between *Simple* and *Advanced*. The first option opens the simple *Transaction creation* dialog (see Section 2.3.3). Choose this option if you only want to quickly transfer some brabocoin to a single address. The second option opens the advanced *Transaction creation* dialog (see Section 2.3.4). Choose this option in order to gain more insight in the transaction data structure. It also allows more complex transactions.
2. With the *Create key pair* button, a new private key is generated, with a corresponding public key and address. The address will be displayed in the list of key pairs (see step 5).
3. The *Save* button saves your wallet. Note that saving your wallet is also done automatically when you close the application.
4. Under *Balance info*, your balance is displayed. It consists of the following parts:
   - The *Confirmed balance* is the user's balance, based only on the transactions that are already mined in a block in the blockchain.
   - *Pending* is the user's additional balance, based on the transactions that are still in the transaction pool.
   - Together, the confirmed balance and pending form the user's spendable balance, the amount of brabocoin the user can actually spend.
   - The *Immature mining reward* consists of the user's mining rewards, which will become spendable after there are 10 blocks mined on top of the user's mined blocks.
5. In the *Key pairs* table, each row indicates a key pair. For each key pair, the address and the balances described above, are displayed. You can right click the address, in order to copy the address itself or the corresponding private key. In the Brabocoin application, private keys can be copied such that signatures can be calculated manually.

2.3.2 Transaction history

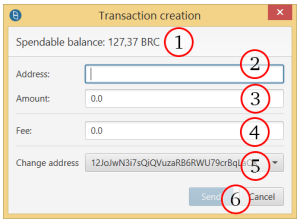
The *Transaction history* page provides a history of all the transactions that involve keys from your wallet.



1. Under *Confirmed transactions*, all your transactions that are included in a block in the main chain are displayed.
2. The *Time received* column displays the time at which the transaction was received (or created).
3. The *Block height* column displays the height of the block that includes the transaction.
4. The *Transaction hash* column displays the hash of the transaction. This is the double SHA-256 hash of the data of the signed transaction.
5. The *Net amount* displays the net amount of brabocoin you spent (if it is red) or received (if it is green) in the transaction.
6. Each row represents a transaction that involves keys from your wallet. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.
7. Under *Unconfirmed transactions*, all your transactions that are not yet included in a block in the main chain, are displayed.
8. The *Time received* column displays the time at which the transaction was received (or created).
9. The *Transaction hash* column displays the hash of the transaction. This is the double SHA-256 hash of the data of the signed transaction.
10. The *Net amount* displays the net amount of brabocoin you spent (if it is red) or received (if it is green) in the transaction.
11. Each row represents a transaction that involves keys from your wallet. When such a row is clicked, the *Transaction detail view* side pane (see Section 2.2.9) will open on the right. This shows detailed information on the transaction that was clicked. If the row is clicked again, the *Transaction detail view* side pane will close.

2.3.3 Transaction creation (simple)

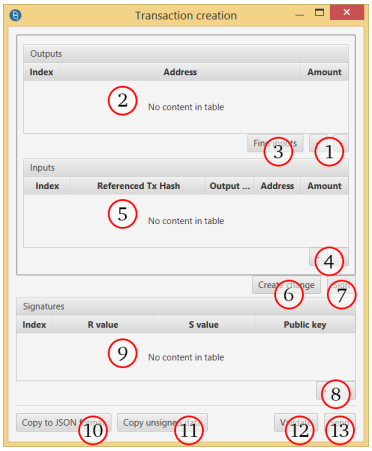
If the *Create transaction* button was clicked in the *Wallet overview* page (see Section 2.3.1), and the option *Simple* was selected, a *Simple transaction creation* dialog will open. In the *Simple transaction creation* dialog, a transaction can be created that transfers brabocoin to a single recipient. These simple transactions can be created to transfer brabocoin quickly.



1. The *Spendable balance* indicates the maximum amount of brabocoin that you can spend.
2. In the *Address* field, the address of the recipient must be filled in, in `Base58Check` format.
3. In the *Amount* field, the amount of brabocoin to be transferred must be filled in.
4. In the *Fee* field, the amount of transaction fees you wish to pay, must be filled in.
5. The *Change address* to which the change of the transaction should be transferred, can be selected from the dropdown.
6. The *Send* and *Cancel* buttons sends the transaction and cancels the transaction creation, respectively.

2.3.4 Transaction creation (advanced)

If the *Create transaction* button was clicked in the *Wallet overview* page (see Section 2.3.1), and the option *Advanced* was selected, an *Advanced transaction creation* dialog will open. In the *Advanced transaction creation* dialog, complex transactions can be created. This dialog also provides more insight in the transaction data structure of Brabocoin.



1. With the plus icon, a new empty output can be added. With the minus icon, a selected output is removed.
2. For each output, the *Address* (in `Base58Check` format) and *Amount* (in brabocoin) must be filled in. The output index is filled in automatically.
3. The *Find inputs* button automatically finds unspent transaction outputs, that you can use to pay the amount specified in your outputs. It is also possible to enter the inputs manually, see steps 4 and 5.
4. With the plus icon, a new empty input can be added. With the minus icon, a selected input is removed.
5. For each input, the *Referenced transaction hash* and *Output index* must be filled in. The remaining fields are filled in automatically, if you are the owner of the referenced unspent transaction outputs.
6. The *Create change* button opens a dialog where you can specify the amount of transaction fee you wish to pay. Also, a change address can be selected. This selection will create an extra output in your transaction, that pays the change of the transaction to the specified address.

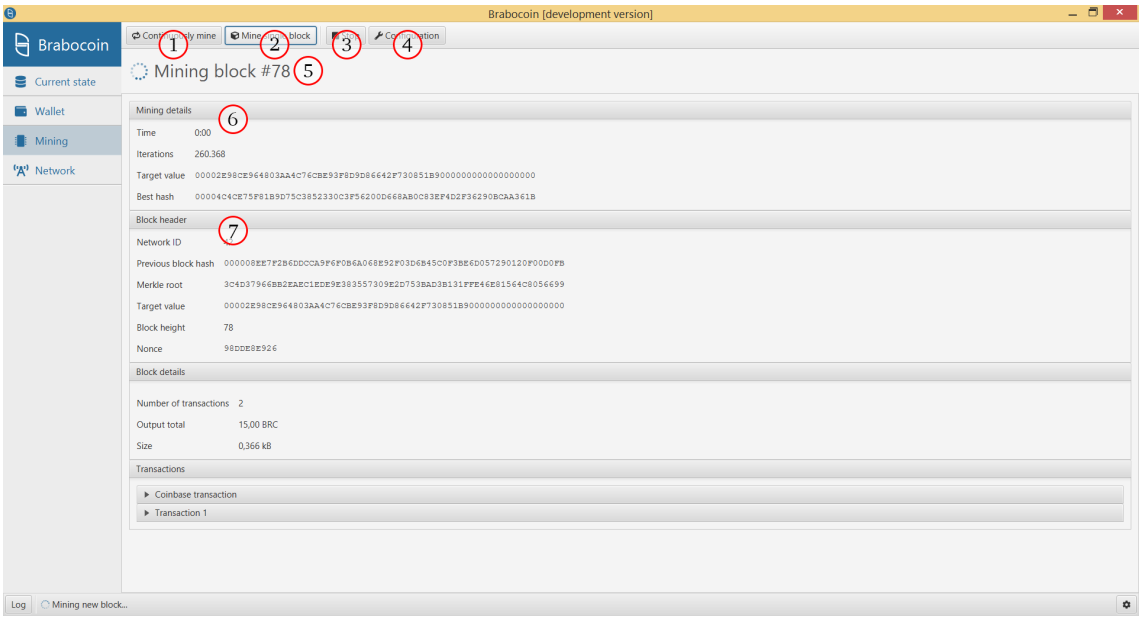
7. The *Sign* button automatically create a signature for each input, provided you own the referenced unspent transaction outputs. It is also possible to enter the signatures manually, see steps 8 and 9.
8. With the plus icon, a new empty signature can be added. With the minus icon, a selected signature is removed.
9. For each signature, the *R* value, *S* value and public key must be specified. The signature index is filled in automatically.
10. The *Copy to JSON format* button copies the entire transaction in JSON format.
11. The *Copy unsigned data* button copies the raw unsigned transaction data, i.e. the transaction without the specified signatures.
12. The *Validate* button opens the *Validation* dialog and validates the transaction (see Section 2.2.11).
13. The *Send* button sends the transaction to your peers.

2.4 Mining

By clicking *Mining* in the navigation menu, as explained in Section 2.1.5, the *Mining* page is opened. On this page, new blocks can be mined on the blockchain.

2.4.1 Mining overview

This page displays all information on the current mining process.



1. The *Continuously mine* button starts a continuous mining process; when a valid block is received with the height at which you are currently mining, it will automatically restart mining a new block with the newly received block as its previous block.
2. The *Mine single block* button starts a mining process that always mines a single block.
3. The *Stop* button will stop any running mining process.
4. The *Configuration* button opens the *Mining configuration* dialog, see Section 2.4.2 for more information.
5. *Mining block #78* indicates the block height of the block that is currently being mined.
6. Under *Mining details*, some details on the current mining process are displayed:
   - The *Time* indicates the elapsed time since you have started mining the current block.
   - *Iterations* indicates the amount of nonce values that have already been tried. For every nonce value, the block hash is calculated and compared to the set target value; if it is smaller, a valid block has been mined.
   - The *Target value* displays the target value that is used to mine this block. This value indicates the difficulty of the proof-of-work algorithm.
   - The *Best hash* displays the lowest block hash that has been found so far.
7. Under *Block header*, the *Block detail view* of the block that is being mined, is displayed. See Section 2.2.8 for more information.

2.4.2 Mining configuration

In the *Mining configuration* dialog, several mining settings can be set. For example, your mining reward address can be specified in this dialog. It is also possible to mine on a fork, instead of mining on the current top block on the main chain, by specifying the hash of the block you wish to mine on.

1. A *Mining reward address* can be selected with the drop-down menu. Note that only addresses included in your wallet can be selected.
2. By default, *Mine on top block* is selected. This means that you always mine on top of the current top block, and restart the mining process if a new top block is received. If you disable *Mine on top block*, this enables you to create forks by mining on any block in the blockchain.
3. By entering a *Parent block hash*, it is possible to mine on a fork, by specifying the block hash of the block you wish to mine on. This is only possible if *Mine on top block* is disabled.
4. The *OK* and *Cancel* buttons save and discard the configuration changes, respectively.

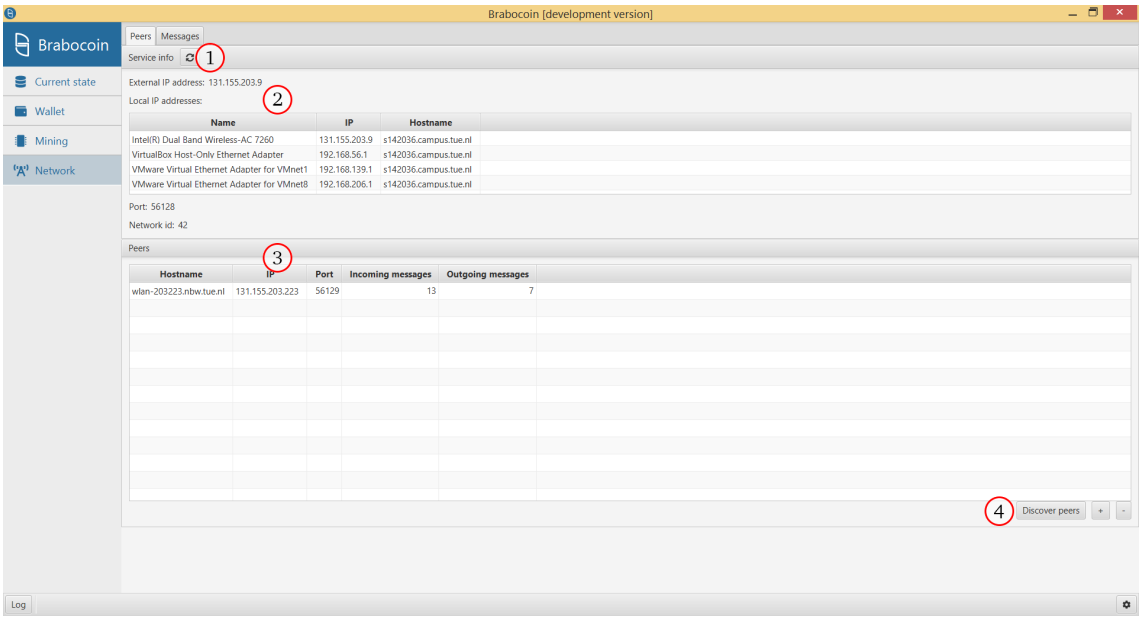
2.5 Network

By clicking *Network* in the navigation menu, as explained in Section 2.1.5, information on your current network is displayed. The following tabs can be opened:

- The *Peers* tab contains information on your current network and a list of your peers.
- The *Messages* tab contains a list of all network messages that you have sent and received.

2.5.1 Peer overview

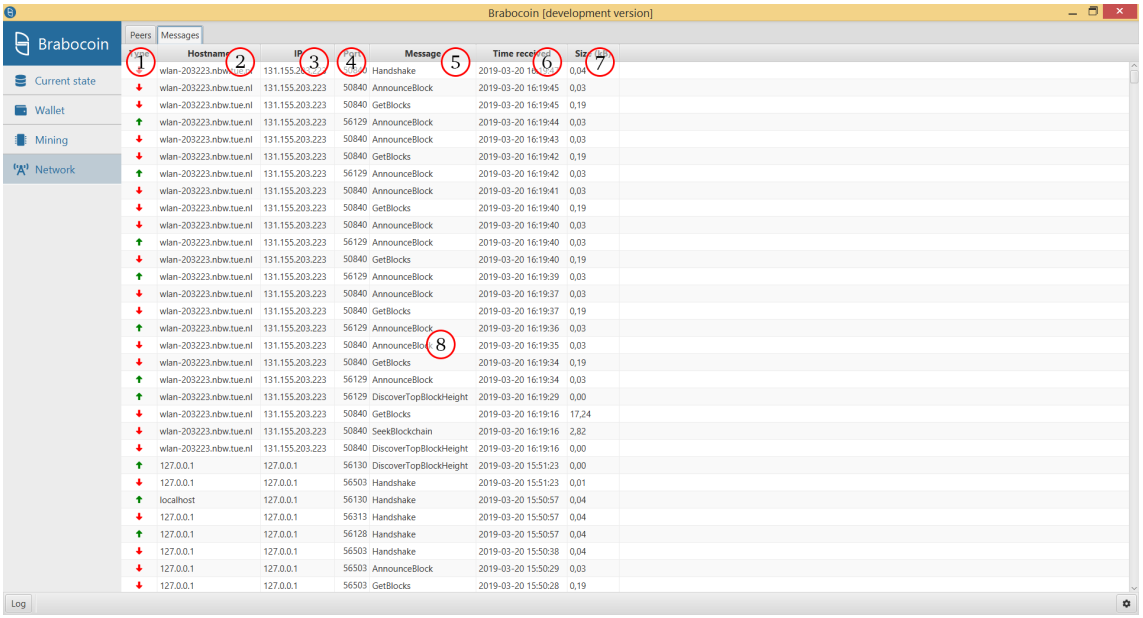
The *Peer overview* page gives an overview of your current network and of your peers. On this page, it is possible to add and remove peers.



1. The *Refresh* button will update the displayed service info.
2. Under *Service info*, your external IP address, local IP addresses, port and network ID are displayed. When a peer is added, its external IP address needs to be entered. By default, the port number used is `56129`. Make sure you have the same network ID as the peers you wish to connect with.
3. Under *Peers*, a list of all your peers is provided. For each peer, their hostname, external IP address and port number are displayed. Also, the amount of incoming messages and the amount of outgoing messages are counted.
4. By clicking the plus icon, the *Peer creation* dialog is opened, which can be used to add a new peer. By clicking the minus icon, a selected peer is removed. By clicking the *Discover peers* icon, the application attempts to find new peers automatically.

2.5.2 Messages

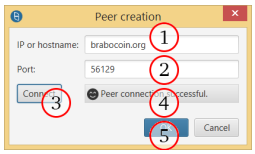
The *Messages* page shows a list of all messages that have been sent to and received from any of your peers.



1. The *Type* is either incoming or outgoing. An incoming message is represented by a red arrow facing down; an outgoing message is represented by a green arrow facing up.
2. The *Hostname* column displays the hostname of the sender or receiver of the message.
3. The *IP* column displays the external IP address of sender or receiver of the message.
4. The *Port* column displays the port number of the sender or receiver of the message.
5. The *Message* column displays which message was sent or received.
6. The *Time received* column displays at which time the message was sent or received.
7. The *Size (kB)* column displays the size of the message in kilobytes.
8. Each row represents a message. When such a row is clicked, the *Message detail view* side pane will open on the right. This shows the request and response of the clicked message in JSON format. It is also possible to open a *Data view* dialog (see Section 2.2.10). If the row is clicked again, the *Message detail view* side pane will close.

2.5.3 Peer creation

In the *Peer creation* dialog, new peers can be added manually.



1. In the *IP or hostname* field, the IP or hostname of the new peer must be entered. For example, you can use the external IP address of your peer, which they can find in their peer overview (see Section 2.5.1).
2. In the *Port* field, the port number of the new peer must be entered. Again, they can find this port number in their peer overview. By default, the port number used is `56129`.
3. With the *Connect* button, an attempt will be made connect with the peer. Note that the peer is not yet added to your list of peers at this point.
4. The status of the connection attempt is displayed next to the *Connect* button.
5. The *OK* and *Cancel* buttons add the peer to your list of peers or close the dialog without adding the peer, respectively.

3 Tutorials

In this section, several tutorial will be provided. Section 3.1 provides some tutorials on the installation of the Brabocoin application. Next, Section 3.2 provides tutorials on the starting of Brabocoin application. In Section 3.3, some general tutorials are provided. The remaining sections describe the functionality of the Brabocoin tool, grouped by the main menu item.

3.1 Installation

3.1.1 Installing the Brabocoin application

In order to install the Brabocoin application, the executable must be downloaded from the Brabocoin website, as described in the procedure.

Windows

1. Go to `https://brabocoin.org/download/`.
2. Click on the installer you want to download. We recommend to use the `Brabocoin-<version>.exe` installer for Windows.
3. Open the file you downloaded.
4. The installation dialog will now open. Follow the instructions of the installation dialog and finally, click *Finish* to exit the installation dialog.

Linux and MacOS

1. Go to `https://brabocoin.org/download/`.
2. Download the `brabocoin-<version>.jar` file.
3. Open a terminal prompt in the folder in which you downloaded the JAR file.
4. Start the brabocoin application by running the command

```
$ java -jar brabocoin-<version>.jar
```

Important
Brabocoin on Linux or MacOS requires an existing Java 8 installation of *at least* update 181 or higher. Brabocoin currently does not run on Java 11 or higher.

Installing the headless version
Brabocoin can also be started in headless mode, where no graphical user interface is provided. It is recommended that headless mode is only used for server installations, as not all functionality is available in headless mode.

- Go to `https://brabocoin.org/download/`.
- Download the `brabocoin-<version>-headless.jar` file.
- Open a terminal prompt in the folder in which you downloaded the JAR file.
- Start the brabocoin application by running the command

```
$ java -jar brabocoin-<version>-headless.jar -p <your wallet password>
```

3.1.2 Installing the Brabocoin calculator

Next to the Brabocoin application, a Brabocoin calculator was developed, which can be used to calculate hashes and perform elliptic curve operations manually. See the integrated help page in the calculator for more information.

Windows

1. Go to `https://www.win.tue.nl/~bdeweger/downloads/BCCv1.0.exe` to download the application.
2. Open the file that you downloaded. The application will now be opened.

Linux and MacOS

1. Go to `https://www.win.tue.nl/~bdeweger/downloads/BCCv1.0.jar` to download the application.
2. Open the file that you downloaded. The application will now be opened.

Info
For more information on how to use the Brabocoin calculator, click on the 'Help' button in the Brabocoin calculator application.

3.2 Starting and opening the application

3.2.1 First-time use

When the Brabocoin application is started for the first time, a password must be created that secures your wallet.

1. Open the Brabocoin application. First, information will be loaded from disk (see Section 2.1.1). Then, the *Password creation* dialog will be opened (see Section 2.1.2).
2. Enter a password in the *Password* field.
3. Repeat the password in the *Confirm password* field. It is essential you remember this password, as you are required to enter it every time you start Brabocoin, in order to unlock your wallet.
4. Click the *Unlock* button. The application will create your wallet and starts synchronizing the blockchain. After the blockchain has been synchronized (see Section 2.1.4), you can use the Brabocoin application.

Important
When you forget your wallet password, there is no way of recovering your wallet.

3.2.2 Starting Brabocoin application

When you start the Brabocoin application at a later point, you already created a wallet the first time you started the application. The application will ask for your password to unlock the wallet.

1. Open the Brabocoin application. First, information will be loaded from disk (see Section 2.1.1). Then, the *Unlock wallet* dialog will be opened (see Section 2.1.3).
2. Enter your password in the *Password* field.
3. Click the *Unlock* button. The application will start synchronizing the blockchain (see Section 2.1.4). After that, you can use the Brabocoin application.

3.2.3 Opening the Brabocoin application

1. Open the Brabocoin calculator executable (`BCCv1.0.exe` or `BCCv1.0.jar`).

3.3 General

In this section, some general tutorials of the Brabocoin application are provided.

3.3.1 Viewing the application log

The application log provides information on what the application is doing.

1. Click the *Log* icon to open the application log (see Section 2.1.6).
2. If you want to change the logging level, click the dropdown after *Show log level*. It you want to close the application log, click the *Log* button again or click the minus icon in the top right corner of the application log.

3.3.2 Changing settings

In the settings menu, several settings of the Brabocoin application can be changed.

1. Click the *Settings* icon to open the *Settings* dialog (see Section 2.1.7).
2. The settings are grouped by *Network*, *Storage* and *Consensus*. Change any setting by editing the form fields.
3. Click the *Save* button. This will display a dialog stating that restarting the application is required.
4. Click the *OK* button to save the changes and close the application. Manually restart the application to apply the changes.

Important
Consensus settings are network-dependent. Make sure you use the same consensus settings as the other nodes in the network. When you change these values, your blocks or transactions might be considered invalid by other nodes in the network.

3.3.3 Copy value

On any displayed page, it is possible to copy any value (that is not in a table) in the following way.

1. Right click the value you want to copy.
2. Click *Select all* to select the value.
3. Right click the selected value.
4. Click *Copy* to copy the value.

3.3.4 Copy value in a table

Some pages show a table with either blocks, transactions or transaction outputs. It is possible to copy any value in a table in the following way.

1. Right click on the value in the table that you want to copy.
2. Click *Copy* to copy the value.

3.4 Current state

In all tutorials in this section, we assume that the *Current state* menu item is opened (see Section 2.2).

3.4.1 Viewing information on the current state

The information on the current state is divided up in several pages. These pages can be opened by clicking one of the tabs. The available tabs are described in Section 2.2.

3.4.2 Viewing and closing detailed block information

The *Blockchain*, *Orphan blocks* and *Recently rejected blocks* pages all display tables of blocks (see Section 2.2.1, 2.2.3 and 2.2.5). In these tables, it is possible to open the *Block detail view*, by clicking on one of the rows of the displayed tables (see Section 2.2.8). This view provides more detailed information on the selected block.

1. Click the block in the table of which you want to have more information. The *Block detail view* will open. Section 2.2.8 describes the information that is displayed on this page.
2. The *Block detail view* can also be closed, by clicking on the currently selected block in the table.

3.4.3 Viewing and closing detailed transaction information

The *Transaction pool*, *Orphan transactions* and *Recently rejected transactions* pages display a table of transactions (see Sections 2.2.2, 2.2.4 and 2.2.6). In these tales, it is possible to open the *Transaction detail view*, by clicking on one of the rows of the displayed tables (see Section 2.2.9). This view provides more detailed infomration on the selected transaction

1. Click the transaction in the table of which you want to have more information. The *Transaction detail view* will open. Section 2.2.9 describes the information that is displayed on this page.
2. The *Transaction detail view* can also be closed, by clicking on the currently selected transaction in the table.

3.4.4 Validating a block

1. Make sure that a *Block detail view* is opened, as described in Section 3.4.2.
2. Click the *Validate* button. If the button is not visible because the *block detail view* is too narrow, click the overflow button (displayed by two right carets) and then click *Validate*.
3. Two types of validation are offered:
    - Click *Quick* for quick validation. Quick validation skips some validation checks, because these checks are expensive to verify once a block is processed and included in the blockchain. Contextual checks, which depend on the state of the blockchain when the validated block was or is connected, are skipped.
    - Click *Complete* for complete validation. All validation checks are executed. Note that when the blockchain is long, complete validation may take a while to finish, since the state of the blockchain must be reconstructed for the contextual checks to be executed.
4. The *Block validation* dialog is opened and validation is started (see Section 2.2.11).

3.4.5 Validating a transaction

1. Make sure that the *Transaction detail view* is opened, as described in Section 3.4.3.
2. Click the *Validate* button. If the button is not visible because the *transaction detail view* is too narrow, click the overflow button (displayed by two right carets) and then click *Validate*.
3. The *Transaction validation* dialog is opened and validation is started (see Section 2.2.11).

3.4.6 Viewing raw block data

If you would like to verify a block hash manually, you need to calculate the hash of the raw block data. This information is available in the *Data view* dialog (see Section 2.2.10).

1. Make sure that the *Block detail view* is opened, as described in Section 3.4.2.
2. Click the *Show data* button. If the button is not visible because the *block detail view* is too narrow, click the overflow button (two right carets) and then click *Show data*.
3. The *Data view* dialog is opened (see Section 2.2.10).

3.4.7 Viewing raw unsigned transaction data

If you would like to manually calculate or verify a signature in a transaction, you need the raw unsigned transaction data. The raw unsigned transaction data contains all transaction data, except the signatures. This information is available in the *Data view* dialog (see Section 2.2.10).

1. Make sure that the *Transaction detail view* is opened, as described in Section 3.4.3.
2. Click the *Show data* button. If the button is not visible because the *transaction detail view* is too narrow, click the overflow button (two right carets) and then click *Show data*.
3. Click *Unsigned transaction*.
4. The *Data view* dialog is opened (see Section 2.2.10).

3.4.8 Viewing raw signed transaction data

If you would like to verify a transaction hash manually, you need to calculate the hash of the raw signed transaction data. This information is available in the *Data view* dialog (see Section 2.2.10).

1. Make sure that the *Transaction detail view* is opened, as described in Section 3.4.3.
2. Click the *Show data* button. If the button is not visible because the *transaction detail view* is too narrow, click the overflow button (two right carets) and then click *Show data*.
3. Click *Signed transaction*.
4. The *Data view* dialog is opened (see Section 2.2.10).

3.4.9 Propagate a block or transaction over the network

When a valid block or transaction is received, it is automatically propagated to your peers. When the block or transaction is invalid, it is not propagated automatically. It is, however, possible to propagate blocks or transactions manually at a later point. It is possible to propagate transactions in the *Transaction pool* and *Recently rejected transactions* pages (see Sections 2.2.2 and 2.2.6), and blocks in the *Recently rejected blocks* page (see Section 2.2.5).

1. Make sure that the *Block* or *Transaction detail view* is opened, as described in Section 3.4.2 or Section 3.4.3, respectively.
2. Click the *Propagate to peers* button. The block or transaction is now propagated to your peers.

3.4.10 Sort the UTXO set

The *UTXO set* page displays a table with all unspent transaction outputs (see Section 2.2.7). By default, it is sorted on block height, then on transaction hash and lastly on block index. It is possible to sort on other fields in the table as well.

1. Click the table header on the column you wish to sort on. If you want it sorted in reverse order, click again. Clicking once again removes the sorting condition on the column.
2. If you also want a secondary column to sort on, hold *Shift* and click in the table header of the second column you want to sort on. You can also sort on additional columns by holding *Shift* and selecting the additional columns.

3.5 Wallet

In all tutorials in this section, we assume that the *Wallet* menu item is opened (see Section 2.3.1).

3.5.1 Creating a transaction

To transfer brabocoin easily, you can create a transaction with the *Simple transaction creation* dialog (see Section 2.3.3). If you want more insight in the transaction data structure or want to create a transaction with multiple outputs, you can create an advanced transaction as described in Section 3.5.2.

1. Click the *Create transaction* button.
2. Click *Simple*.
3. Fill in the address of the recipient (in `Base58Check` format) in the *Address* field.
4. Fill in the amount of brabocoin to be transferred in the *Amount* field. Note that amounts can be specified in up to two decimal places.
5. Fill in the transaction fee in the *Fee* field.
6. Select a change address by clicking the dropdown and then clicking the address on which you want to receive the change of the transaction.
7. Click the *Save* button to validate the transaction.
8. If the transaction is valid, a dialog will be displayed confirming whether you want to send this transaction to your peers. Click the *OK* button to send the transaction to your peers or *Cancel* to stop the procedure. If the transaction is invalid, a dialog will be displayed warning you the transaction is invalid. Click the *OK* button to send the transaction to your peers anyway or *Cancel* to stop the procedure.

3.5.2 Creating a transaction (advanced)

If you want more insight in the transaction data structure or want to create a transaction with multiple outputs, you can create an advanced transaction with the *Advanced transaction creation* dialog (see Section 2.3.4).

1. Click the *Create transaction* button.
2. Click *Advanced*.
3. Click the plus icon below the *Outputs* table to add a new empty output.
4. Fill in the *Address* of the recipient (in Base58Check format) and the *Amount* (in brabocoin).
5. Repeat steps 3 and 4 for all additional outputs you wish to add.
6. Click the *Find inputs* button if you want to find unspent transaction outputs automatically. These are used as inputs for the created transaction and will appear in the inputs table. It is also possible to enter the inputs manually with the following steps:
   (a) Click the plus icon below the *Inputs* table to add a new empty input.
   (b) Fill in the the *Referenced transaction hash* and *Output index* of the referenced unspent transaction output.
   (c) Repeat steps 6a and 6b for all additional inputs you wish to add.
7. Click the *Create change* button.
8. Fill in the transaction fee in the *Fee* field.
9. Select a change address, by clicking the dropdown and then clicking the address on which you want to receive the change of the transaction.
10. Click the *OK* button.
11. Click the *Sign* button if you want to create the signatures of the transaction automatically. It is also possible to enter the signatures manually with the following steps:

(a) Click the plus icon below the *Signatures* table to add a new empty signature.

(b) Fill in the *R* value, the *S* value and the *Public key*.

(c) Repeat steps 11a and 11b for every additional signature you wish to add. The amount of signatures must be equal to the amount of inputs, as each signature signs one input.

12. Click the *Send* button to send the transaction.
13. If the transaction is valid, a dialog will be displayed confirming whether you want to send this transaction. Click the *OK* button to send the transaction to your peers, or *Cancel* to stop the procedure. If the transaction is invalid, a dialog will be displayed stating this. Click the *OK* button to send the transaction to your peers anyway, or *Cancel* to stop the procedure.

3.5.3 Creating a new key pair

The Brabocoin wallet keeps track of all your addresses. The first address is created automatically when the application is first started (see Section 3.2.1). Additional addresses must be created manually.

1. Click the *Create key pair* button. This will add a key pair to your list of key pairs.

Info
Generating a new key pair might take several seconds.

3.5.4 Saving the wallet

The wallet is automatically saved when you close the Brabocoin application. It is also possible to save your wallet manually.

1. Click the *Save* button. This will save the wallet.

Info
We recommend saving your wallet manually every time after you generate a new key pair.

3.5.5 Copy an address from your key pairs

1. In the *Key pairs* table, right click on the row that contains the key pair of which you want to copy the address.
2. Click *Copy address* to copy the address.

3.5.6 Copy a private key from your key pairs

In a normal cryptocurrency wallet, you never want to copy a private key, for security reasons. However, private keys can be used to create signatures manually or to allow others to try to forge a signature and thus, steal your funds. Therefore, copying a private key is made available in the Brabocoin application.

1. In the *Key pairs* table, right click on the row that contains the key pair of which you want to copy the private key.
2. Click *Copy private key* to copy the private key.

Important
If you want to protect your funds, never share your private key with others! Anyone with the private key has full control of the funds paid to the corresponding address.

3.5.7 Viewing transaction information in transaction history

In the *Transaction history* page is opened, the most important information of each transaction is displayed (see Section 2.3.2). It is possible to view additional transaction information by opening the

Transaction detail view (see Section 2.2.9).

1. Click the transaction in the table of which you want to have more information. The *Transaction detail view* will open. Section 2.2.9 describes what information is displayed on this page. Sections 3.3.3, 3.4.5, 3.4.7 and 3.4.8 describe more detailed tutorials about the *Transaction detail view*.
2. The *Transaction detail view* can also be closed, by clicking on the currently selected transaction in the table.

3.6 Mining

In all tutorials in this section, we assume that the *Mining* menu item is opened (see Section 2.4.1).

3.6.1 Continuously mine

In *Continuously mine* mode, once a new block is mined, the application automatically starts to mine on top of the newly mined block.

1. Click the *Continuously mine* button. The application is now mining and will not stop unless you click the *Stop* button or close the application.

3.6.2 Mining a single block

When *Mining a single block*, once you have mined a new block, the mining procedure stops.

1. Click the *Mine single block* button. The application will now mine a single block. If another block is received from the network before the block is mined, the application will stop mining immediately.

Info
By default, the block is mined on top of the current top block in the blockchain. See the tutorial in Section 3.6.5 if you want to mine on top of another block.

3.6.3 Stop mining

This procedure will stop any running mining procedure.

1. Click the *Stop* button. The application will now stop mining.

3.6.4 Changing the mining reward address

Using this tutorial, you can specify which address you wish to receive your mining rewards and transaction fees on.

1. Click the *Configuration* button. This opens the *Mining configuration* dialog (see Section 2.4.2).
2. Select the mining reward address by clicking the dropdown and then clicking the address on which you want to receive the mining reward.

3.6.5 Changing the previous block hash

Using this tutorial, you can specify which block will be included in your block as parent. When you start a mining procedure, a new block will be mined on this specified block.

1. Click the *Configuration* button. This opens the *Mining configuration* dialog (see Section 2.4.2).
2. Uncheck the *Mine on top block* checkbox.
3. Enter the block hash of the block on which you want to mine. When mining a new block, this block will be used as previous block hash.

Info
Check the *Mine on top block* checkbox again if you want to automatically mine on the current top block in the blockchain.

3.7 Network

In all tutorials in this section, we assume that the *Network* menu item is opened (see Section 2.5.1).

3.7.1 Adding a peer

To connect to another node in the Brabocoin network, you need to add them as a peer.

1. Click the plus icon in the bottom right. This opens a *Peer creation* dialog (see Section 2.5.3).
2. In the *IP or hostname* field, the IP or hostname of the new peer must be filled in. For example, you can use the peer's external IP address. They can find their external IP address on their peer overview page.
3. In the *Port* field, the port number of the new peer must be filled in. This port is stated on their peer overview page and is `56129` by default.
4. Click the *Connect* button. The status of the connection attempt will be displayed next to this button. If the connection failed, check whether the *IP or hostname* and *Port* fields are filled in correctly and try again.
5. Click the *OK* button to add the peer.

3.7.2 Viewing message data

If the *Messages* page is opened (see Section 2.5.2), all incoming and outgoing messages between you and your peers are displayed. It is possible to view the raw message data of a message that was sent by clicking on the message.

1. Click the message in the table of which you want to see the raw message data. The request and response of the messages will be displayed in JSON format.
2. To view the request or response of the message in hexadecimal format, click the *Show data* button above the message. This will open a *Data view* dialog ((see Section 2.2.10) of the selected message.