SmartCash Documentation

Release latest

strophy

Apr 20, 2021

First Steps

1	Contents			3
	1.1	What is	SmartCash?	3
		1.1.1	SmartCash Videos	3
		1.1.2	Brochure/Whitepaper	4
		1.1.3	Roadmap	4
	1.2	Feature	S	4
		1.2.1	Specifications	4
		1.2.2	Smartreward	4
		1.2.3	Smartnode	4
		1.2.4	InstantPay	5
		1.2.5	Smartvote	5
		1.2.6	Smartcard	6
		1.2.7	SmartBand	6
		1.2.8	Hive Structuring Teams	7
		1.2.9	Keccak Hash Algorithm	7
		1.2.10	Emission Rate	7
		1.2.11	Fees	8
	1.3	How To	Buy	9
		1.3.1	Exchanges	0
		1.3.2	Instant exchanges	.3
		1.3.3	Over the Counter	.4
		1.3.4	ATMs 1	5
	1.4	Safety .		5
		1.4.1	Scams	.6
		1.4.2	Ponzi Schemes	6
	1.5	Links a	nd Information	.7
		1.5.1	Links 1	.7
		1.5.2	Tools	21
		1.5.3	Mobile Apps	2
		1.5.4	Glossary	2
	1.6	Wallets		29
		1.6.1	NodeClient Wallet	:9
		1.6.2	Electrum Wallet	57
		1.6.3	Android Wallet(deprecated) 9	95
		1.6.4	iOS Wallet(deprecated)	95
		1.6.5	Smartcard Wallet	15

	1.6.6	Paper Wallet
	1.6.7	Hardware Wallets
	1.6.8	Third Party Wallets
	1.6.9	Web Wallets(deprecated)
	1.6.10	Wallet Guides
1.7	Earning	, Saving and Spending
	1.7.1	Earning 117
	1.7.2	Saving
	1.7.3	Spending
	1.7.4	Tax
1.8	Getting	Started
	1.8.1	Payment Processors
	1.8.2	Installation Examples
1.9	Admini	strative Processes
	1.9.1	Onboarding Process
	1.9.2	Promoting SmartCash
	1.9.3	Currency Conversion
	1.9.4	Legal considerations
1.10	Technic	al Guides
	1.10.1	SmartCash Wallet Integration
	1.10.2	Integration Notes
	1.10.3	API Services
	1.10.4	SDK Resources
	1.10.5	InstantPay
	1.10.6	Vending Machines
	1.10.7	Price Tickers
	1.10.8	QR Codes
1.11	SmartH	ive
	1.11.1	SmartHive Teams
	1.11.2	Discussion
	1.11.3	SmartVote
1.12	SmartR	ewards
	1.12.1	SmartRewards FAQ'S
1.13	SmartN	odes
	1.13.1	Understanding Smartnodes
	1.13.2	Hosting Services
	1.13.3	Setup
	1.13.4	Maintenance
	1.13.5	Advanced Topics
1.14	Mining	
	1.14.1	SmartNodes vs. Mining
	1.14.2	Mining Pools
1.15	Develop	pers
	1.15.1	Translating SmartCash
	1.15.2	Compiling SmartCash NodeClient
	1.15.3	Testnet
	1.15.4	Insight API Installation
	1.15.5	Version History
1.16	Marketi	ng
	1.16.1	Design Materials
	1.16.2	Business Templates
1.17	Legal .	
	1.17.1	Smartcash Deemed Not a Security Under Singapore Law
	1.17.2	How the Law Applies to SmartCash

1.17.3 ATM & Fiat Compliance	165
------------------------------	-----

HAS NOT BEEN OFFICIAL RELEASED YET



SmartCash is a peer-to-peer digital currency that can be used without having to trust a third party, which makes it possible to be your own bank. The name reflects the aim: By focusing on being ease to use even for people who aren't crypto geeks, on being merchant friendly and with a self-funded unique governance model SmartCash aims to be the cryptocurrency of everyone's choice when it comes to payments, SmartCash currently also offers instant transactions (*InstantPay*) and operates a self-governing and self-funding model that enables the SmartCash network to pay individuals and businesses for work that adds value to the network. This *smarthive and budgeting system*.

If you are new to cryptocurrencies, the most important change to understand is that transactions occur directly between two parties without any central authority to facilitate the transaction. This also means that you are responsible for your own security - there is no bank or credit card company to reverse a transaction if your funds are stolen or lost. In this sense, it is similar to cash or gold, but cryptocurrency can be spent locally and internationally with equal ease, if you are confident you are sending funds to the right destination. For these reasons, the SmartCash documentation has a strong focus on safety and understanding the concepts and features that drive the SmartCash ecosystem. The videos, links and documentation below can help you get started, or use the table of contents on the left to find a specific topic of interest.

CHAPTER 1

Contents

1.1 What is SmartCash?

SmartCash is a peer-to-peer digital currency that can be used without having to trust a third party. By focusing on being ease to use even for people who aren't crypto geeks, on being merchant friendly. The SmartCash network features *instant transaction confirmation*, double spend protection, a *self-governing, self-funding model* and a *roadmap*. While SmartCash is based on Bitcoin and compatible with many key components of the Bitcoin ecosystem.

This section of the documentation describes these and many more key features that set SmartCash apart in the blockchain economy.

The videos, links and documentation collected here can help you get started, or use the table of contents on the left to find a specific topic of interest. New users may be interested in getting started with an appropriate *wallet*, learning about *how to buy SmartCash* and *where to spend SmartCash*, learning about *safety* or joining one of the many *SmartCash community sites*.

1.1.1 SmartCash Videos

SmartCash Explained

SmartCash Explained is a six-part video series. It explains SmartCash from a beginner's level up to descriptions of the more advanced features. Please note some content in this video are out of date like smartreward payments times, Fungibility and Privacy features.

SmartCard & SmartPay

SmartCard & SmartPay is an nine-part video series. It covers unique features and functions of SmartCard & SmartPay only available on the SmartCash network, and is available in English,

1.1.2 Brochure/Whitepaper

The SmartCash Brochure/Whitepaper describes the unique value proposition and key innovations in SmartCash from an academic and theoretical perspective. It is a living document maintained on https://smartcash.cc/brochure/, receiving ongoing updates and frequent community translations as new features are implemented.

Latest whitepaper and official translations

1.1.3 Roadmap

The SmartCash Roadmap sets out delivery milestones for future releases of SmartCash and includes specific technical details describing how the development team plans to realese each challenge. Like the Brochure/Whitepaper, it is versioned and maintained on .

SmartCash Roadmap

1.2 Features

1.2.1 Specifications

- First block mined at 11:24 PM, July 11, 2017
- No premine
- Keccak hashing algorithm, GPU mining available
- 55 seconds block time, 4MB blocks, ~232 transactions per second
- BlockReward = 5000x(143,500/BlockHeight)(max.5000)
- BRNDF difficulty adjustment algorithm
- 5 Billion total coin supply
- Instant transactions using instantpay
- Decentralized Governance By Blockchain allows SMART owners to vote on budget proposals and decisions that affect SmartCash

1.2.2 Smartreward

SmartRewards are a method of distributing block rewards among SmartCash addresses holding at least 1000 Smart-Cash and has no outgoing transaction over the period of 47500 Blocks in order to disperse voting power and coin production among the wider community. This allows for a more decentralized creation of block rewards. Payouts occur 200 Blocks after the cycle ends and every second block 1000 addresses receive rewards. SmartNodes will keep track on which addresses are eligible and which are not.

1.2.3 Smartnode

In addition to traditional Proof of Work (PoW) rewards for mining SmartCash, users are also rewarded for running and maintaining special servers called smartnodes. Thanks to this innovative two tier network, SmartCash can offer innovative features in a trustless and decentralized way. Smartnodes are used to power instantpay, smartreward, and the other features. Users are rewarded for running smartnodes; 10% of the block reward is allocated to pay the smartnodes network. You can view practical guides on all topics relating to smartnodes *here*.

Smartnodes enable the following services:

- **InstantPay** Allows for SmartCash transactions to be locked in about a second. No risk of double spending a transaction, so the receiver can trust that transaction immediately.
- **Smartrewards** are calculated by the SmartNodes to allow for distribution to be handled automatically by the block rewards.
- Electrum Servers Servers used for Electrum wallets.
- SmartCard SDK Build the SmartCash payment card that can handle world usage in replacement of card schemes.

SmartNodes are required to have

- 100000 SMART collateral
- A computer for running a local wallet
- A dedicated VPS server running 24/7 with the following specifications:
- 2 GB RAM 20 GB drive Static IPV4 Address per node

SmartNodes cost money and effort to host so they are paid a percentage of the block reward as an incentive. This tool shows a live calculation of smartnode earnings. These rewards decrease by time, but the rising value of Smart may offset these reductions.

1.2.4 InstantPay

Traditional decentralized cryptocurrencies must wait for certain period of time for enough blocks to pass to ensure that a transaction is both irreversible and not an attempt to double-spend money which has already been spent elsewhere. This process is time-consuming, and may take anywhere from 15 minutes to one hour for the widely accepted number of six blocks to accumulate. Other cryptocurrencies achieve faster transaction confirmation time by centralizing authority on the network to various degrees.

SmartCash suffers from neither of these limitations thanks to its second-layer network of smartnodes. Smartnodes can be called upon to form voting quorums to check whether or not a submitted transaction is valid. If it is valid, the smartnodes "lock" the inputs for the transaction and broadcast this information to the network, effectively promising that the transaction will be included in subsequently mined blocks and not allowing any other spending of these inputs during the confirmation time period.

InstantPay technology will allow for cryptocurrencies such as SmartCash to compete with nearly instantaneous transaction systems such as credit cards for point-of-sale situations while not relying on a centralized authority. Widespread vendor acceptance of SmartCash and InstantPay could revolutionize cryptocurrency by shortening the delay in confirmation of transactions from as long as an hour (with Bitcoin) to as little as a few seconds.

SmartCash InstantPay Faster Than Visa or Mastercard

1.2.5 Smartvote

All Smartcash users deserve a voice, from day #1. That is why Smartcash is pioneering in the crypto-governance space and was built around Smarthive, the platform that lets all users vote on the direction of the project. The one feature that sets SmartCash apart is the SmartHive governance model and how the community can vote to fund the best projects with block rewards stored in the SmartHive Project Treasury. The idea is that the SmartHive Governance Portal engages community members and other 3rd parties interested in growing SmartCash into a global currency. Everyone can create a proposal that helps SmartCash in some way. Then every SmartCash holder also can participate in the voting process and help decide if its worth getting funded or not by the SmartHive Project Treasury. All users have the ability to vote; 1 SMART equals 1 vote. Proposals are open to vote for 14 days, with one day extensions if the vote outcome changes at the end of that period and payments will be made in accordance with the delivered

milestones. Proposals open to vote can be found at: vote.smartcash.cc Basically, each SmartCash coin is worth 1 vote, to keep it simple.

1.2.6 Smartcard

Smartcard (*Getting Started*) Debit and credit card payments have become the dominant way for many to carry out most, if not all, of their daily transactions. Adoption of card payments became increasingly simple for small businesses and now quick and affordable smartphone-powered solutions have widespread adoption. SmartCard emerged offering to solve many of the problems traditional payment systems are facing such as long settlement times, chargebacks, and high fees. SmartCard is a blockchain-based solution that aims to improve on the digital payment card model. Our main goals are to streamline crypto transactions and make them practical for use in real-life payment scenarios. At the same time we aim to drastically reduce fees and confirmation time frames over those found in current solutions. The SmartCard makes it possible to pay with SmartCash using a physical card where no mobile wallet or phone internet access is required. Since the SmartCard doesn't use a bank or a credit card processor, transactions are not subject to any restrictions. Merchants will have a point of sale (POS) app, so they transmit transactions for you but can't initiate the payment without your permission. For further information about how the system works and how both merchants and customers will benefit from using the SmartCard and SmartCash POS App please visit: smartcash.cc/smartcard/

Advantages

- No 3rd Party. Instant SmartCash payments to merchants anywhere in the world with 0% fees.
- Pay only with your SmartCard QR. You don't need to carry your phone or have internet connection.
- Have all the SmartCards you need under your account and they don't expire.
- Print anywhere you love the most. Tattoos included.
- Purchase online or offline from your favorite SmartMerchants.
- Convenience that all family understands.
- Open Source code with world wide contributors.

Security

- Protections such as card on/off toggle, country whitelisting, daily spend limit and PIN reset.
- Zero knowledge of SmartCard private keys.
- Never exposes private keys to the internet.
- Private keys are generated and encrypted in the browser and printed by you.
- Payment authorisation and signing is done in the browser and SmartPay merchant application.
- A decentralised payment system that gives you full control of your funds.

1.2.7 SmartBand

The SmartBand is a partnership between Stratum, ATAR, and SmartCash. ATAR launched a wristband which is linked to an account where people in Brazil can deposit fiat and pay with it for their daily needs using NFC technology built in the band at over 2.5 million merchants. The partnership with Stratum, an international exchange, makes it possible to link your exchange SmartCash wallet with your SmartBand to load the ATAR account with SmartCash. Stratum automatically converts SmartCash to the local currency required for purchases. Once a payment is done, you will receive a notification on your phone for each transaction. With over 85% of all point of sale terminals in Brazil

accepting the ATAR Band, this pioneering cooperation allows SmartCash to be accepted broadly by merchants with no additional steps, technology, or training required.

1.2.8 Hive Structuring Teams

The Hive Structuring Teams (HST) will lay the groundwork for the project and set the framework for the project's evolution. Initially, the Hive Structuring Teams will operate three separate departments in charge of vital project areas, such as development, outreach and support. Each of these three departments will end up employing people. As teams grow larger, bigger than eight members, more teams will be created, each with its own budget and team coordinator, operating autonomously without a central command and control structure in place.

The Hive Structuring Teams will consist of people directly involved in the project on a day to day basis, while the overall SmartHive will aim to enrol and engage community members and other 3rd parties interested in growing SmartCash into a global worldwide currency.

1.2.9 Keccak Hash Algorithm

Keccak (pronounced as "ketchak") also known as SHA-3 (Secure Hash Algorithm 3). It is the latest generation secure hashing algorithm released by NIST (National Institutes of Standards and Technology) in 2012. Keccak is a family of cryptographic sponge functions and is designed as an alternative to SHA-256 – An algorithm used by Bitcoin and various other crypto currencies. Compared to SHA-256, Keccak (SHA-3) is much faster and is more secure. If you are looking for Keccak Whitepaper then head to this link. If you are looking for specifications and more in-depth information on Keccak encryption algorithm then check this and this. Information on mining with Keccak can be found in the *Mining* section of this documentation.

Advantages of Keccak

Keccak is a SHA-3 hashing algorithm and as it is a member of SHA family it isn't 100% ASIC resistant. In fact it is ASIC friendly but as far as we know there isn't an ASIC currently for this algorithm.

1.2.10 Emission Rate

Cryptocurrencies such as SmartCash and Bitcoin are created through a cryptographically difficult process known as mining. Mining involves repeatedly solving *hash algorithms*. Once discovered, the miner is permitted to create new units of the currency. This is known as the block reward. To ensure that the currency is not subject to endless inflation, the block reward is reduced at regular intervals, as shown in this calculation. Graphing this data results in a curve showing total coins in circulation, known as the coin emission rate.

While SmartCash is based on Bitcoin, it significantly modifies the coin emission rate to offer a smoother reduction in coin emission over time. While Bitcoin reduces the coin emission rate by 50% every 4 years.

SmartCash first year is approximately then

717,565,000+998,173,000=1,715,738,000 or 34.31% of the maximum supply.

Year 2 generates an additional 500 Million SmartCash, bringing the total supply to about 2.2 Billion out of the total 5 Billion, or 44%.

Year 3 generates even less about 291.7 Million more SmartCash, for a total supply of about 50%, or 2.5Billion out of 5 Billion. The remainder is spread over the next 121 years.



Fig. 1: SmartCahs coin emission rate 125 Years

Total coin emission

Bitcoin's total coin emission can be calculated as the sum of a geometric series, with the total emission approaching (but never reaching) 21,000,000 BTC. This will continue until 2140, but the mining reward reduces so quickly that 99% of all bitcoin will be in circulation by 2036, and 99.9% by 2048.

SmartCash's total coin emission. SmartCash will emit coins for approximately 124.3 years before a full year of mining creates less than 1 SMART. After 2141 no more SMART will be created.

Read more here.

Block reward allocation

Unlike Bitcoin, which allocates 100% of the block reward to miners, SmartCash allocates:

5% Mining	5% Mining Reward	
10%	Smartnodes	
15%	SmartRewards	
4%	x6 Hive Structuring Teams	
46%	SmartHive Project Treasury	

1.2.11 Fees

Transactions on the SmartCash network are recorded in blocks on the blockchain. The size of each transaction is measured in bytes, but there is not necessarily a correlation between high value transactions and the number of bytes required to process the transaction. Instead, transaction size is affected by how many input and output addresses are involved, since more data must be written in the block to store this information. Each new block is generated by a miner, who is paid for completing the work to generate the block with a block reward. In order to prevent the network

from being filled with spam transactions, the size of each block is artificially limited. As transaction volume increases, the space in each block becomes a scarce commodity. Because miners are not obliged to include any transaction in the blocks they produce, once blocks are full, a voluntary transaction fee can be included as an incentive to the miner to process the transaction. Most wallets include a small fee by default, although some miners will process transactions even if no fee is included.

instantpay fee

InstantPay transactions, which operate on a different and mandatory fee schedule. SmartCash introduced InstantPay autolocks, which causes smartnodes to automatically attempt to lock any transaction with 4 or fewer inputs—which are referred to as "simple" transactions—and removes the additional fee for InstantPay. The fee schedule for SmartCash as of December 2018 is as follows:

Transaction type	Recommended fee	Per unit	
Standard transaction	.00001 SMART	Per kB of transaction data	
InstantPay autolock	.00001 SMART	Per kB of transaction data	
InstantPay	.0001 SMART	Per transaction input	

As an example, a standard and relatively simple transaction on the SmartCash network with one input, one output and a possible change address typically fits in the range of 200 - 400 bytes. Assuming a price of US\$1 per SMART, the fee falls in the range of \$0.000002 - \$0.000004, or 1/5000th of a cent. Processing a simple transaction using InstantPay at the same price is free of charge, while more complex InstantPay transactions may cost around 0.01-0.02 cents per transaction, depending on the number of inputs. These fees apply regardless of the SmartCash or dollar value of the transaction itself.

1.3 How To Buy

SmartCash can be purchased and sold in several ways, each with different advantages and disadvantages. The following options are available:

- *Exchanges* are one of the most popular ways to trade cryptocurrency. A wide range of exchanges exist, each offering slightly different features. Some serve different markets, some are in direct competition, some have cheaper fees, and some are subject to more or less strict regulatory requirements. Most exchanges are centralized, meaning they are operated by a single company, which may be obliged by the laws of the jurisdiction in which it operates to collect data on its customers. Others are decentralized, but as a result have higher escrow requirements since you are dealing peer-to-peer instead of with a trusted entity. Exchanges can be broadly broken down into two categories: exchanges which accept national currency (fiat money) and exchanges which deal in cryptocurrencies only. For safety, exchanges should not be used as wallets. Exchanges are for trading, not for savings.
- *Instant exchanges* perform a similar function to normal exchanges, but without the requirement to log in. They effectively convert one currency to another, with some limits on the amount to be exchanged and usually at a less advantageous rate. Others may even offer to sell cryptocurrency as a credit card purchase.
- *Over the counter* exchanges have recently appeared to facilitate sale of SmartCash directly from a company to the individual at a specified price, or peer-to-peer between individuals at a negotiated price. Volume may be limited compared to exchanges, but these services are usually much easier to use. More advanced peer-to-peer sites offer escrow services for a fee to prevent cheating during the sale between two parties who have never met.
- *ATMs* accepting card and cash payments in return for crypto are widely available. Mapping services can show the specific location of these machines, or you can even set one up at your own business and earn a percentage of sales.

DISCLAIMER: This list is provided for informational purposes only. Services listed here have not been evaluated or endorsed by SmartCash and no guarantees are made as to the accuracy of this information. Please exercise discretion when using third-party services.

1.3.1 Exchanges

Cryptocurrency exchanges exist to convert national currency, also known as fiat money, into cryptocurrency. Many exchanges do not accept fiat money, and exchange between various cryptocurrencies only. Trades are handled on markets, and trades are created between pairs of currencies, identified by their ticker codes. SmartCash is widely accepted on exchanges and many pairs exist against both fiat money and cryptocurrency. This means it is possible to exchange EUR for SMART, or SMART for BTC, for example. The volume traded on an exchange provides a good indication of how quickly a buy or sell order you place will be filled. This section introduces some of the most popular exchanges for trading SMART.

Marketplace comparison websites

Coingecko

https://www.coingecko.com/en/coins/smartcash/trading_exchanges#panel

CoinGecko provides a fundamental analysis of the crypto market. In addition to tracking price, volume and market capitalization, CoinGecko tracks community growth, open-source code development, major events and on-chain metrics.

CoinMarketCap

https://coinmarketcap.com/currencies/smartcash/#markets

CoinMarketCap lists all cryptocurrencies by their market capitalization. Clicking one of these currencies allows you to view price charts, and clicking Markets allows you to view the markets available and the trading pairs they offer.

List of exchanges

The exchanges listed here are for informational purposes only and do not indicate endorsement or affiliation with any particular platform.

Exmo

https://exmo.com

Exmo is a UK-registered exchange offering fiat deposits in USD, EUR, USD and UAH (Ukrainian Hryvnia). SMARTCASH trading pairs exist for BTC, USD and RUB.



E/// EXMO

HitBTC



CoinMarketCap



10

https://hitbtc.com

HitBTC offers facilities to major investors to credit USD, EUR and GBP, as well as BTC, ETH and USDT trading pairs against SMART for normal users.

Safe.trade

https://safe.trade/trading/bcsmart

SafeTrade is a new, innovative trading platform for cryptocurrencies from the folks who brought you SafeCoin.

Coinbene

The World's Leading Trustworthy Crypto Assets Platform | BitcoinEthereumBlockchain Asset Trading Platform https://www.coinbene.com

Stex

Cryptotrading platform https://app.stex.com/en/basic-trade/pair/BTC/SMART/

eBitcoinics

The largest Bitcoin & CryptoCurrency educational platform in Africa & your best eCurrency exchange platform https://www.ebitcoinics.com/

pungo.cash

Buy and sell Cryptocurrency

https://pungo.cash

Swapzone

https://swapzone.io/?to=smart

Please be aware: some of the partner services have floating exchange rates and AML/KYC procedures. Swapzone users are responsible for their actions based on the information stated here. Swapzone is not responsible



peer2cash

STEX







coinbene

for the risks or losses that might occur as a result of taking an action based on this information any way, regardless of the user's interpretation of this data. It is the user's responsibility to conduct due diligence of the listed projects prior to taking action. It is recommended to consult the community and form an opinion before making a transaction.

Localbitcoincash

Buy Bitcoin Cash, Sell Bitcoin Cash, Trade Bitcoin Cash Locally https://localbitcoincash.org/index.asp

Bitcoinp2p

Compre bitcoin no P2P, não precisa de documentos, é anônimo e não tem fila de espera. https://bitcoinp2p.com.br/index.html

Stratum

Stratum

https://stratum.hk/

Bitprime

BitPrime operates a secure and compliant platform for easy retail trading in New Zealand.

https://www.bitprime.co.nz/product/smartcash-smart/

Finexbox

Finexbox.

https://www.finexbox.com/

PARS

https://parsxc.com/









BITCOIN^{P2P}

stratum

coinBR







Pars is an international Iranian user to user cryptocurrency exchange. The exchange has a large potential audience (about 90 Million population country) and aims to become widely known, improving its services and adding new interesting cryptocurrencies, thus increasing the interest of users in the exchange.



Deliondex

https://dex.delion.online/

Decentralized CryptoCurrency Exchange offer trading pairs for SmartCash.

1.3.2 Instant exchanges



https://changelly.com

Changelly is a broker service offering a range of cryptocurrency, including SmartCash, for instant exchange against other cryptocurrencies without needing to create an account. Be sure to check the fees and rates before purchasing.



🖓 changelly

Swapzone

https://swapzone.io/?to=smart

Please be aware: some of the partner services have floating exchange rates and AML/KYC procedures. Swapzone users are responsible for their actions based on the information stated here. Swapzone is not responsible for the risks or losses that might occur as a result of taking an action based on this information any way, regardless of the user's interpretation of this data. It is the user's responsibility to conduct due diligence of the listed projects prior to taking action. It is recommended to consult the community and form an opinion before making a transaction.

CoinSwitch

https://coinswitch.co

CoinSwitch is a crypto to crypto exchange aggregate with more than 300 different coins and tokens listed. Also offers purchases through credit/debit cards.

SimpleSwap

https://simpleswap.io/

SimpleSwap is a simple and easy-to-use platform for cryptocurrency exchanges that works without registration and limits



SimpleSwap

Swapspace.co

https://swapspace.co

SwapSpace is a cryptocurrency exchanger aggregator that will save your time.

Swap4.me

https://swap4.me

Swap4.me.

Stealthex

https://stealthex.io Anonymous and limitless cryptocurrency exchange Free from providing personal data Your privacy is our top concern

BitTorro

https://bittorro.io BitTorro is a crypto to crypto, wallet to wallet, no-custody, swap exchange service. Here you can instant swap one cryptocurrency for another in minutes. As a privacy focused platform, your personal data are safe and never shared.

1.3.3 Over the Counter

eBitcoinics

http://www.ebitcoinics.com

eBitcoinics is a cryptocurrency exchange and education platform for the African market. SmartCash is available for exchange against Nigerian Naira (NGN) and Ghanaian Cedi (GHS).

Bitcoin-Avenue

http://bitcoin-avenue.com/

The French Bitcoin Avenue, a brick and mortar cryptocurrency shop.





Bitcoin ävenue...



SWAP4.ME

Stealthe





Changelly

https://changelly.com

Changelly is a popular instantaneous crypto to crypto exchange platform with more than 100 different coins and tokens listed. Also offers purchases via credit/debit cards.



Stratum CoinBR

https://coinbr.io

CoinBR is a Brazil-based cryptocurrency company offering a variety of services including an exchange, mining, bill payment, point-of-sale, and more. Smartcash is available for purchase at over 13,000 locations around Brazil.

1.3.4 ATMs

ATMs are a popular method of buying cryptocurrency at businesses to encourage adoption and spending in these currencies. A number of ATMs support SmartCash, and the mapping services listed on this page can help you find one near you. It is also possible to operate your own ATM to sell SmartCash on-site at your business - simply contact the companies listed on this page.



General Bytes

https://www.generalbytes.com

General Bytes offers a range of two-way cash ATM and Point of Sale solutions integrating SmartCash.

BitugaX

bitugax 🕟

https://www.bitugax.com/

You can buy SmartCash at BitugaX ATM in Coimbra, Portugal!

1.4 Safety

If you are new to cryptocurrencies, the most important change to understand in comparison with the traditional banking system is that transactions occur **directly between two parties without any central authority** to facilitate the transaction. This also means that **you are responsible for your own security** - there is no bank or credit card company to reverse a transaction if your funds are stolen or lost. If you forget or lose your wallet file, recovery phrase or PIN, you will permanently and irrevocably lose access to your funds.

SmartCash is a project born out of the desire to create an easy to use, fast, and secure cryptocurrency that can support everyday use such as business payments and daily transactions. In this sense, it is similar to cash or gold, but cryptocurrency can be spent locally and internationally with equal ease, if you are confident you are sending funds to the right destination. For these reasons, the SmartCash documentation has a strong focus on safety and understanding the concepts and features that drive the SmartCash ecosystem. A few general safety guidelines:

- Do not trust any online service or person because they sound or look reputable. Always use an escrow service if you are buying peer-to- peer.
- Store your SmartCash on a *hardware wallet* if possible. If not, then store your coins in the official *SmartCash NodeClient Wallet* or the official *SmartCash Electrum Wallet*.
- Do not use exchanges as wallets. Exchanges are for trading, not for savings.
- Mobile wallets should be used for day-to-day purchases, but do not keep large amounts of funds in them. Transfer funds as necessary.

A list of known scams, fake wallets and Ponzi or pyramid schemes can be seen below. Do NOT trust them.

1.4.1 Scams

There are many "fake" pages on the internet attempting to trick users into sending SmartCash or other cryptocurrencies or "open a wallet". Other scams include selling fake mining hardware, fake SmartCash or altcoins with a similar name, and Ponzi schemes (see below). Beware of fake Twitter accounts impersonating SmartCash! The official Twitter account is: https://twitter.com/scashofficial

Please report these and any others scams you encounter as follows:

- 1. Report phishing and scams to Google: https://www.google.com/safebrowsing/report_phish
- 2. Look up the registrar of the domain and send a complaint: https://www.whois.com/whois
- 3. Report phishing to Netcraft: https://www.netcraft.com
- 4. Report scams to the BadBitcoin Project: http://www.badbitcoin.org
- 5. If in doubt, use Crypto Scam Checker to see if already report and report there as well: https://fried.com/ crypto-scam-checker

Feel free to report any new scams you find on the forum: https://forum.smartcash.cc

1.4.2 Ponzi Schemes

A Ponzi scheme, Pyramid scheme or Multi-level marketing are a fraudulent investment operations where the operator provides fabricated reports and generates returns for older investors through revenue paid by new investors. More and more users must constantly join the scheme in order for it to continue operation, with ever greater numbers of people losing money to the originators of the scheme.

- What is a Pyramid Scheme?
- How to spot a Ponzi Scheme
- · BehindMLM News and blog about Ponzi schemes

If you encounter a Ponzi scheme, follow the same reporting steps as above for scam websites!

List of known Ponzi schemes (there are many more - stay vigilant!):

OneCoin

- http://themerkle.com/dr-ruja-flees-sinking-ship-as-regulators-crack-down-on-onecoin/
- http://siliconangle.com/blog/2016/09/29/dodgy-cryptocurrency-onecoin-under-police-investigation-accused-of-being-a-police-invest
- https://cointelegraph.com/news/one-coin-much-scam-onecoin-exposed-as-global-mlm-ponzi-scheme
- http://www.makemoneyexpert.com/online/network-marketing/reviews/onecoin/

• https://pageone.ng/2016/11/05/beware-onecoin-ponzi-scheme/

SwissCoin

- http://behindmlm.com/mlm-reviews/swisscoin-review-25-to-15000-eur-ponzi-points-investment/
- http://ethanvanderbuilt.com/2017/01/26/swisscoin-scam-warning/
- https://news.bitcoin.com/dissecting-swisscoin-cryptocurrency-ponzi-horizon/

The Billion Coin

- https://steemit.com/news/@rahmat/review-the-billion-coin-ponzi-scheme
- https://coins.newbium.com/post/728-scam-alert-the-billion-coins-scam-ponzi-scheme
- https://bitcointalk.org/index.php?topic=1592288.0

Sustaincoin

• http://www.scamvoid.com/check/sustaincoin.com

E-Dinar

- http://behindmlm.com/mlm-reviews/e-dinar-review-edr-unit-ponzi-points-cryptocurrency/
- https://www.scam.com/showthread.php?714218-E-dinar-coin
- https://bitcointalk.org/index.php?topic=1569896.0

DasCoin

- http://behindmlm.com/mlm-reviews/coin-leaders-review-dascoin-is-a-onecoin-ponzi-points-clone/
- https://bitcointalk.org/index.php?topic=1636850.0

BitConnect

- https://www.reddit.com/r/Bitconnect/comments/76fa9k/bitconnect_investigated_as_a_ponzi_scheme/
- https://www.youtube.com/watch?v=6fujWfmgRJU
- http://www.binaryoptionsarmy.com/2017/11/bitconnect-scam-review/
- https://satoshiwatch.com/hall-of-shame/bitconnect-coin/

HashOcean

• http://themerkle.com/bitcoin-scam-risk-warning-hashocean/

CryptoDouble

• http://themerkle.com/bitcoin-hyip-ponzi-scheme-alert-coindouble/

1.5 Links and Information

1.5.1 Links

Official sites

- Website: https://smartcash.cc/
- Wallets/Clients: Windows/Mac/Linux/Android https://smartcash.cc/wallets/
- Web Wallet: https://wallet.smartcash.cc/

- Paper Wallet: https://paperwallet.smartcash.cc/
- 3rd Party Wallets:
 - https://atomicwallet.io/
 - https://www.coinomi.com/
 - https://smart.edge.app/
- SmartCard: https://smartcash.cc/smartcard/
- SmartPay App:
 - https://business.smartcash.cc/
 - https://play.google.com/store/apps/details?id=cc.smartcash.pos
- SmartHive Project Voting:
 - https://vote.smartcash.cc/
 - https://forum.smartcash.cc/c/smarthive/pre-proposal-discussion
- SmartCash Media:
 - https://smartcash.cc/social-media/
 - https://twitter.com/scashofficial
 - https://www.reddit.com/r/smartcash/
 - http://discord.smartcash.cc/
 - http://t.me/smartcashchat
 - https://www.minds.com/smartcashoutreach1
 - http://forum.smartcash.cc/
 - http://youtube.smartcash.cc/
 - https://medium.com/@smartcashnews
 - https://www.instagram.com/scashofficial/
 - https://www.facebook.com/scashofficial/
 - https://gab.ai/smartcash
- Block Explorers:
 - https://insight.smartcash.cc/
 - https://explorer3.smartcash.cc/
 - https://smart.ccore.online/
- Statistics & Calculators:
 - https://explorer2.smartcash.cc/ext/getmoneysupply
 - https://stats.smartcash.cc/
 - https://smartcash.cc/smartnodes/#toggle-id-1
- GitHub:
 - https://github.com/SmartCash/Core-Smart
 - https://github.com/SmartCash/electrum-smart

• Roadmap:

- https://smartcash.cc/roadmap/
- SmartBand:
 - https://wearatar.com/band
- Brochure / Whitepaper:
 - Website: https://smartcash.cc/brochure/

Table 1: Brochures

LanguageDownload				
English	PDF_SmartCash_brochure_en-1.2.8			
Russian	PDF_SmartCash_brochure_ru_1.2			
German	PDF_SmartCash_brochure_DE_1.4			
Spanish	PDF_SmartCash_brochure_es_1.2.4			
PortugueseDF_SmartCash_brochure_ptbr_1.2				
Japanes	ePDF_SmartCash_brochure_ja_1.2			
Vietnam	eseDF_SmartCash_brochure_vi_1.2			

• Exchanges:

- Full List: https://smartcash.cc/get-smartcash/
- Exmo: https://exmo.com/
- Changelly: https://changelly.com/
- HitBTC: https://hitbtc.com/

• SmartNodes:

- https://smartcash.cc/smartnodes
- https://github.com/SmartCash/smartnode
- https://smartcash.freshdesk.com/solution/articles/35000047674
- https://smartcash.freshdesk.com/helpdesk/attachments/35037201968
- http://t.me/smartnodemonitorbot
- Charts:
 - https://bravenewcoin.com/smartcash
 - https://coinmarketcap.com/currencies/smartcash/
 - https://www.coingecko.com/en/coins/smartcash

Community sites

- Portfolio Tools:
 - http://blockfolio.com/
 - https://getdelta.io/#features
 - https://cointracking.info/
 - HODL iOS: https://itunes.apple.com/us/app/hodl-real-time-cryptocurrency/id1253668876
 - HODL Android: https://play.google.com/store/apps/details?id=com.github.hintofbasil.hodl

- Event Calendar: https://smartcash.cc/event-directory/ https://coindar.org/en/coin/smartcash
- Mining Guide: https://steemit.com/smartcash/@keyd0s/smartcash-mining-guide
- Mining Pools:
 - https://us-mine.smartcash.cc/
 - https://eu-mine.smartcash.cc/
 - https://asia-mine.smartcash.cc/
 - https://nlpool.nl/
 - http://antminepool.com/
 - https://blake2s.com/
- Mining Programs:
 - Claymore Dual Miner: https://bitcointalk.org/index.php?topic=1433925.0
 - NVIDIA: https://github.com/tpruvot/ccminer/releases and https://github.com/KlausT/ccminer/ releases
 - AMD: https://github.com/genesismining/sgminer-gm/releases
- Mining Stats:
 - https://whattomine.com/coins/197-smart-keccak
 - https://stats.smartcash.cc/mining
- SmartCash News: https://medium.com/@smartcashnews
- SmartCash Newsletter: https://smartcash.cc/signup

Forums

• https://forum.smartcash.cc

Charts

- https://bravenewcoin.com/smartcash
- https://coinmarketcap.com/currencies/smartcash/
- https://www.coingecko.com/en/coins/smartcash

Social media

- https://smartcash.cc/social-media/
- https://twitter.com/scashofficial
- https://www.reddit.com/r/smartcash/
- http://discord.smartcash.cc/
- http://t.me/smartcashchat
- https://www.minds.com/smartcashoutreach1
- http://forum.smartcash.cc/

- http://youtube.smartcash.cc/
- https://medium.com/@smartcashnews
- https://www.instagram.com/scashofficial/
- https://www.facebook.com/scashofficial/
- https://gab.ai/smartcash

News

- https://smartcash.cc/news/
- https://steemit.com/@smartcashnews/
- https://www.mycryptopedia.com/smartcash-masternode-explained/

Blogs

Wikipedia

Inactive

1.5.2 Tools

Block explorers, statistics and visualizations

- Block Explorers:
 - https://insight.smartcash.cc/
 - https://explorer3.smartcash.cc/
 - https://smart.ccore.online/
- Statistics & Calculators:
 - https://explorer2.smartcash.cc/ext/getmoneysupply
 - https://stats.smartcash.cc/
 - https://smartcash.cc/smartnodes/#toggle-id-1

Treasury tools

https://vote.smartcash.cc/

Smartnode management

• https://steemit.com/smartcash/@dustinface/v2-0-smartnodemonitor-update

Price monitoring and statistics

- https://www.coingecko.com/en/coins/smartcash
- https://coinmarketcap.com/currencies/smartcash

SmartCash Community project

- https://smartie.shop/wp/
- https://stats.smartcash.cc/
- https://smartcashradio.libsyn.com/

1.5.3 Mobile Apps

iOS

- https://itunes.apple.com/us/app/smartcash-wallet/id1438215565?mt=8
- https://itunes.apple.com/us/app/coinomi-wallet/id1333588809?mt=8
- https://itunes.apple.com/us/app/edge-bitcoin-wallet/id1344400091?mt=8

Android

- https://play.google.com/store/apps/details?id=cc.smartcash.wallet
- https://play.google.com/store/apps/details?id=com.coinomi.wallet
- https://play.google.com/store/apps/details?id=co.edgesecure.app
- https://play.google.com/store/apps/details?id=cloud.peer2.pungo_wallet

1.5.4 Glossary

- **51% Attack** A condition in which more than half the computing power on a cryptocurrency network is controlled by a single miner or group of miners. That amount of power theoretically makes them the authority on the network. This means that every client on the network believes the attacker's hashed transaction block.
- Address A SmartCash address is used to *Send/Receive a Payment* on the SmartCash network. It contains a string of alphanumeric characters, but can also be represented as a scannable QR code. A SmartCash address is also the public key in the pair of keys used by SmartCash holders to digitally sign transactions (see Public key).
- **Algorithm** In mathematics and computer science, an algorithm is a self-contained step-by-step set of operations to be performed. Algorithms perform calculation, data processing, and/or automated reasoning tasks.
- Altcoin Since Bitcoin was the first cryptocurrency and has the largest market capitalization, it is considered as the reference. An altcoin, or alternative coin, is any cryptocurrency other than Bitcoin.
- **AML** Anti-Money Laundering techniques are used to stop people from making illegally obtained funds appear as though they have been earned legally. AML mechanisms can be legal or technical in nature. Regulators frequently apply AML techniques to SmartCash exchanges.
- **API** In computer programming, an application programming interface (API) is a set of routines, protocols, and tools for building software and applications.

An API expresses a software component in terms of its operations, inputs, outputs, and underlying types, defining functionalities that are independent of their respective implementations, which allows definitions and implementations to vary without compromising the interface. A good API makes it easier to develop a program by providing all the building blocks, which are then put together by the programmer.

- **ATM / BTM** A SmartCash ATM is a physical machine that allows a customer to buy SmartCash with cash. There are many manufacturers, some of which enable users to sell SmartCash for cash. They are also sometimes called 'BTMs' or 'SmartCash AVMS.' SmartCash is supported on several *ATMs*.
- Backlog Backlog generally refers to an accumulation over time of work waiting to be done or orders to be fulfilled.
- **Backup** The process of making copies of a computer file to ensure its integrity in case of loss, theft, or damage. SmartCash allows users to *make backup copies* of their digital wallets. This protects against losing one's money in the event of a computer crashing or losing one's mobile device. This would be the equivalent of being able to backup the cash in your wallet, so that if you lost it, you could restore the cash from a backup.
- **Bitcoin 2.0** This is a term explaining the next new level of Bitcoin projects which started as a fork of Bitcoin but extended their code into the next level of Blockchain Projects (Smart Contracts, Decentralised Voting,....)
- **Blockchain** A blockchain is a distributed database that maintains a continuously-growing list of data records hardened against tampering and revision. It consists of data structure blocks which exclusively hold data in initial blockchain implementations, and both data and programs in some of the more recent implementations with each block holding batches of individual transactions and the results of any blockchain executables. Each block contains a timestamp and information linking it to a previous block.
- **Blocks** Transactions on the Blockchain are collected in "blocks" which record and confirm when and in what sequence transactions enter and are logged in the block chain. Blocks are created by users known as "miners" who use specialized software or equipment designed specifically to create blocks.
- SmartHive Project Treasury The development of SmartCash and the SmartCash ecosystem is self-funded by the network. Each time a block is discovered, 5% of the block reward goes to miners and 10% goes to smartnodes. 46% is withheld by the network and used to fund projects that are approved by the community. This process is known as *SmartHive*. For a fee, anybody can submit a proposal to the network, and will be paid directly by the SmartHive Project Treasury if approved by the community. The SmartHive Project Treasury System is sometimes called the Treasury System; the two terms are interchangeable.
- **SmartMining** This is a powerful method of mitigating 51% mining attacks, which are associated with double spending.
- **Cloud Wallet** Third parties that will store your SmartCash on their servers for you, so that you can access your funds from any device connected to the internet. If their website is hacked or if their servers are damaged, you run the risk of losing your SmartCash. Any online wallets should be secured with strong passphrases and 2FA. You cannot make backup copies of your online wallet, because you do not have access to the private keys. We do not recommend that you store large quantities of funds in online wallets.
- **Coinbase transaction** The first transaction in a block. Always created by a miner, it includes a single input which constitutes the block reward.
- **Cold Storage** A method of generating and storing private keys completely offline. One could use a desktop or laptop computer disconnected from the internet, a dedicated hardware wallet, a USB stick, or a *paper wallet*.
- **Confirm(ed) Transaction** When a SmartCash transaction is made, a miner must verify that the transaction is valid. When the inputs and outputs are verified, the transaction is included in a block in the blockchain. The transaction can then be considered complete and irreversible. The confirmation number increases as more blocks are added to the blockchain.
- **Confirmation Number** The number of confirmations for a specific SmartCash transaction. Zero confirmations means that the **transaction is unconfirmed**. One confirmation means that the transaction is included in the latest block in the blockchain. Two confirmations means the transaction is included in two blocks, three confirmations for three blocks, and so on. The probability of a transaction being reversed (double spent) diminishes exponentially with every block and subsequent confirmation. Six confirmations is usually considered "safe" and irreversible.
- **Confirmed Transactions** Transactions that are processed by miners and considered irreversible, usually after six confirmations. In the case of InstantSend, funds can be considered irreversible after a few seconds, but must still be written to the blockchain (and thus "confirmed").

- **CPU** A central processing unit (CPU) is the electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logical, control and input/output (I/O) operations specified by the instructions. The term has been used in the computer industry at least since the early 1960s. Traditionally, the term "CPU" refers to a processor, more specifically to its processing unit and control unit (CU), distinguishing these core elements of a computer from external components such as main memory and I/O circuitry.
- **Cryptocurrency** A cryptocurrency (or crypto currency or crypto-currency) is a medium of exchange using cryptography to secure the transactions and to control the creation of new units.
- **Cryptography** Cryptography or cryptology (from Greek $\kappa\rho\upsilon\pi\tau$ kryptós, "hidden, secret"; and $\gamma\rho\phi\epsilon\iota\nu$ graphein, "writing," or $-\lambda o\gamma \alpha$ -logia, "study," respectively) is the practice and study of techniques for secure communication in the presence of third parties called adversaries. More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages; various aspects in information security such as data confidentiality, data integrity, authentication, and non- repudiation are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.
- **NodeClient Wallet** The *SmartCash NodeClient Wallet* (known also as the QT wallet) is the "official" SmartCash wallet that is compiled by the SmartCash Team and allows InstantPay. The NodeClient wallet will download the entire blockchain and serve it over the internet to any peers who request it.
- **DDoS** A distributed denial of service attack uses large numbers of computers under an attacker's control to drain the resources of a central target. They often send small amounts of network traffic across the Internet to tie up computing and bandwidth resources at the target, which prevents it from providing services to legitimate users. SmartCash exchanges have sometimes been hit with DDoS attacks.
- **Decentralized** Decentralized computing is the allocation of resources, both hardware and software, to each individual workstation or office location. In contrast, centralized computing exists when the majority of functions are carried out or obtained from a remote centralized location. Decentralized computing is a trend in modern-day business environments. This is the opposite of centralized computing, which was prevalent during the early days of computers. A decentralized computer system has many benefits over a conventional centralized network. Desktop computers have advanced so rapidly that their potential performance far exceeds the requirements of most business applications. This results in most desktop computers remaining nearly idle most of the time. A decentralized system can use the potential of these systems to maximize efficiency. However, it is debatable whether these networks increase overall effectiveness.
- **Desktop Wallet** A wallet is a piece of software that stores your SmartCash. There are many different wallet options, but it is imperative to choose a secure one. We recommend any of the following: *SmartCash NodeClient Wallet* / *SmartCash Electrum Wallet | Hardware Wallets*
- **Difficulty** This number determines how difficult it is to hash a new block. It is related to the maximum allowed number in a given numerical portion of a transaction block's hash. The lower the number, the more difficult it is to produce a hash value that fits it. Difficulty varies based on the amount of computing power used by miners on the SmartCash network. If large numbers of miners leave a network, the difficulty would decrease. SmartCash's increasing popularity and the availability of specialized ASIC miners have caused the difficulty to increase over time.
- Digital Wallet See *this link* for full documentation on wallets.

A digital wallet is similar to a physical wallet except that it is used to hold **digital currency**. A SmartCash wallet holds your private keys, which allow you to spend your SmartCash. You are also able to make backups of your wallet in order to ensure that you never lose access to your SmartCash. Digital wallets can exist in many different forms and on many devices:

• Desktop Wallet (*SmartCash Electrum Wallet*, *SmartCash NodeClient Wallet*): Wallet programs that you install on a laptop or desktop computer. You are solely responsible for protecting the wallet file and the

private keys it contains. Make backup copies of your wallet files to ensure that you don't lose access to your funds.

- **Mobile Wallet** (*Android*, *iOS*): These wallets can be downloaded through Google Play or Apple (iTunes) App Stores. Mobile wallets allow you to use SmartCash on-the-go by scanning a QR code to send payment. Make backup copies of your mobile wallet files to ensure that you don't lose access to your funds. Due to security issues with mobile phones, it is advised that you don't store large amounts of funds on these wallets.
- **Online/Cloud/Web Wallet** (*Coinomi, Webwallet*): Third parties that will store your SmartCash on their servers for you or provide an interface to access your SmartCash with you providing the keys, so that you can access your SmartCash from any device connected to the internet. If their website is hacked or if their servers are damaged, you run the risk of losing your SmartCash. Any online wallets should be secured with strong passphrases and 2FA. You cannot make backup copies of your online wallet, because you do not have access to the private keys. We strongly urge that you NEVER store large amounts of SmartCash in any cryptocurrency exchange.
- Hardware Wallets (*Trezor*, Ledger, Nano): A hardware wallet is a specialized, tamper-proof, hardware device that stores your private keys. This device is able to sign transactions with your private key without being connected to the internet. However, you must have an internet connection to send the transaction to the SmartCash network. This allows your private keys to be accessed easily while still keeping them securely protected. This is widely regarded to be the safest form of storage for your SmartCash.
- Offline/Cold Storage (*Paper wallet*): A special wallet that is created offline and is never exposed to the internet. Accomplished by using software to generate a public and private key offline and then recording the generated keys. They keys can be printed out on paper or even laser-etched in metal. Copies can be made and stored in a personal safe or bank deposit box. This is an extremely secure way to store SmartCash. There is no risk of using software wallet files, which can become corrupt, or web wallets, which can be hacked. NOTE: USB sticks are not safe for long-term (multi-year) storage because they degrade over time.
- **Digital Signature** A digital signature is a mathematical mechanism that allows someone to prove their identity or ownership of a digital asset. When your digital wallet signs a transaction with the appropriate private key, the whole network can see that the signature matches the address of the SmartCash being spent, without the need to reveal the private key to the network. You can also digitally sign messages using your private key, to prove for instance that you are the owner of a certain SmartCash address.
- **Electrum Wallet** *SmartCash Electrum Wallet* is a lightweight wallet that does not require you to download or sync the entire blockchain, making the wallet lighter and faster.
- **Encryption** In cryptography, encryption is the process of encoding messages or information in such a way that only authorized parties can read it. Encrypted messages which are intercepted by a third-party are indecipherable gibberish without the private key. In an encryption scheme, the *plaintext* message is encrypted using an encryption algorithm, generating *ciphertext* that can only be read if decrypted by the intended recipient. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. Increases in computing power have "broken" many past encryption algorithms, but a well-designed modern system such as AES-256 is considered essentially "uncrackable."

Escrow Services An escrow is:

- a contractual arrangement in which a third party receives and disburses money or documents for the primary transacting parties, with the disbursement dependent on conditions agreed to by the transacting parties; or
- an account established by a broker for holding funds on behalf of the broker's principal or some other person until the consummation or termination of a transaction; or
- a trust account held in the borrower's name to pay obligations such as property taxes and insurance premiums.

A trusted escrow service is often used when purchasing cryptocurrency or other goods/services over the internet. Both the buyer and seller will choose a trusted third-party, the seller will send the item (or currency) to the escrow agent, and the buyer will send the purchasing funds to the escrow agent as well. Once the escrow agent is satisfied that both parties have satisfied the terms of the agreement, he/she will forward the funds and the product (or currency) being purchased to the appropriate party.

- **Exchange** The current price of one SmartCash compared to the price of other currencies, like the US dollar, Yen, Euro, or Bitcoin. Because most trading volume takes place on the BTC/SMART markets, price is often quoted in fractions of a bitcoin. An excellent site for following the exchange rate of SmartCash is CoinMarketCap. Businesses wishing to reduce the risk of holding a volatile digital currency can avoid that risk altogether by having a payment processor do an instant exchange at the time of each transaction.
- **Faucet** Faucets are a reward system, in the form of a website or app, that dispenses rewards in the form of a microsmart or Duff, which is a hundredth of a millionth SmartCash, for visitors to claim in exchange for completing a captcha or task as described by the website.
- Fiat Gateway Fiat money has been defined variously as:
 - Any money declared by a government to be legal tender.
 - State-issued money which is neither convertible by law to any other thing, nor fixed in value in terms of any objective standard.
 - Intrinsically valueless money used as money because of government decree.

Examples include the US dollar, the Euro, the Yen, and so forth.

- **Fintech** Financial technology, also known as FinTech, is an economic industry composed of companies that use technology to make financial services more efficient. Financial technology companies are generally startups trying to make financial processes more efficient or eliminate middle- men. Recently many fintech companies have begun utilizing blockchain technology, which is the same technology that underpins SmartCash and Bitcoin.
- **Fork** When the blockchain diverges or splits, with some clients recognizing one version of the blockchain as valid, and other clients believing that a different version of the blockchain is valid. Most forks resolve themselves without causing any problems, because the longest blockchain is always considered to be valid. In time, one version of the blockchain will usually "win" and become universally recognized as valid. Forks can, however, be extremely dangerous and should be avoided if possible.

Forking is most likely to occur during software updates to the network.

Full Nodes Any SmartCash client that is serving a full version of the blockchain to peers. This can be a user running a SmartCash NodeClient wallet on his/her desktop, or it could be a *smartnodes*. Full nodes promote decentralization by allowing any user to double check the validity of the blockchain.

Fungible Every unit of the currency is worth the same as any other unit.

Genesis Block The very first block in the block chain.

- **GPU** A graphics processing unit (GPU), also occasionally called visual processing unit (VPU), is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing, and their highly parallel structure makes them more efficient than general- purpose CPUs for algorithms where the processing of large blocks of data is done in parallel. In a personal computer, a GPU can be present on a video card, or it can be embedded on the motherboard or in certain CPUs on the CPU die. Certain cryptocurrencies use mining algorithms which are most efficiently run on GPUs.
- Hardware Wallet *Hardware wallets* are among the safest type of wallet for storing your SmartCash. Your private key is protected inside a piece of hardware, and is never exposed to the internet. You are still able to sign transactions as normal, making it both safe and convenient.

- **Hash** A mathematical process that takes a variable amount of data and produces a shorter, fixed-length output. A hashing function has two important characteristics. First, it is mathematically difficult to work out what the original input was by looking at the output. Second, changing even the tiniest part of the input will produce an entirely different output.
- Hashrate The number of hashes that can be performed by a SmartCash miner in a given period of time (usually a second).
- Insight Blockchain information server used to power block explorers and respond to transaction queries.
- **InstantPay** *InstantSend* technology uses the smartnode network to "lock" transaction inputs, preventing SmartCash from being double-spent. Unlike Bitcoin, where it takes an hour or longer for transactions to fully confirm, transactions using InstantPay are "locked" and irreversible after only a few seconds.
- **Liquidity** The ability to buy and sell an asset easily, with pricing that stays roughly similar between trades. A suitably large community of buyers and sellers is important for liquidity. The result of an illiquid market is price volatility, and the inability to easily determine the value of an asset.
- Smartnode A *smartnodes* is special type of full node that performs services for the network and is paid a portion of the block reward. Smartnode require proof of ownership of 100000 SMART.

Smartnodes serve as the second tier of the SmartCash network, and power InstantPay, Smartrewards.

- Mining *Miners* process transactions on the SmartCash network and publish them on the blockchain. As a reward for doing this, miners are paid 5% of the block reward.
- Mobile Wallet These are wallets available on mobile devices (iOS + Android).
- **MultiSig** Multi-signature addresses provide additional security by requiring multiple people to sign a transaction with their private key before the transaction can be sent. For example, in 2 of 3 multisig, two out of three possible signatories have to sign a transaction for it to be processed. Multi-signature addresses are commonly used by exchanges and other organizations that are in possession of large sums of cryptocurrency, since it makes theft much more difficult.
- **Node** A node is any device running SmartCash wallet software. Full nodes are software clients that have downloaded the entire blockchain and serve it to other clients on SmartCash's peer-to-peer network.
- **OTC** Over the counter (OTC) trades are trades that occur off exchanges. In an OTC trade, a buyer and seller trade with each other directly, or through an intermediary. OTC trading is useful when a person wants to either buy or sell a large amount of cryptocurrency and is afraid that a large buy or sell order will move the price (called "slippage").
- **P2P** Peer-to-peer. Decentralized interactions that happen between at least two parties in a highly interconnected network. An alternative system to a 'hub-and-spoke' arrangement, in which all participants in a transaction deal with each other through a single mediation point.
- **Paper Wallet** *Paper wallets* are offline wallets, printed on paper for safety. If properly secured and stored they are considered the safest way to store cryptocurrency.
- **Privacy** Privacy is the ability of an individual or group to seclude themselves, or information about themselves, and thereby express themselves selectively. The boundaries and content of what is considered private differ among cultures and individuals, but share common themes. When something is private to a person, it usually means that something is inherently special or sensitive to them. The domain of privacy partially overlaps security (confidentiality), which can include the concepts of appropriate use, as well as protection of information.
- **Private Key** A private key is a long alphanumeric passcode that allows SmartCash to be spent. Every SmartCash wallet contains one or more private keys which are saved in the wallet file. The private keys are mathematically related to all SmartCash addresses generated for the wallet. Because the private key is the "ticket" that allows someone to spend SmartCash, it is important that these are kept secure and secret.

- **Proof of Service PoSe** Consensus mechanism used in SmartCash to verify that a smartnodes has provided uninterrupted service meeting a minimum quality level to the network. Maintaining this service allows a smartnodes to enter and move up through the global list and eventually into the selection pool to receive payment.
- **Proof of Stake PoS** Consensus mechanism that relies on ownership of a cryptocurrency to maintain the blockchain. In Proof of Stake systems, each owner of the currency can use their wallet to "stake," and there's a small chance that they will be chosen to create the next block and add it to the chain. In this way consensus is maintained across all nodes. Proof of Stake saves electricity and does not require specialized computer hardware. It does however suffer from several pitfalls, including the "nothing at stake" problem. Since no electricity is consumed, in the event of an attack it is actually beneficial for Proof of Stake nodes to "vote" to accept both the legitimate chain and the attacker's chain.
- **Proof of Work PoW** Consensus mechanism that keeps all nodes honest by requiring computational power to be expended in order to create new blocks. Miners must use expensive equipment and burn electricity to add blocks to the blockchain. Without a consensus mechanism of some sort, any node could add blocks to the chain and the network's nodes would never agree on which chain was valid.
- **Public Key** The public key is derived from the private key but is not secret and can be revealed to anybody. When a private key is used to sign messages, the public key is used to verify that the signature is valid.
- **Pump and dump** Inflating the value of a financial asset that has been produced or acquired cheaply, often using aggressive publicity and misleading statements. The publicity causes others to acquire the asset, forcing up its value. When the value is high enough, the perpetrator sells their assets, cashing in and flooding the market, which causes the value to crash. This is particularly common in markets with low liquidity, such as some altcoins.
- **Quorum** Group of smartnodes signing or voting on some action, with the formation of the group determined by some determiniation algorithm.
- **QR Code** A two-dimensional graphical block containing a monochromatic pattern representing a sequence of data. QR codes are designed to be scanned by cameras, including those found in mobile phones, and are frequently used to encode SmartCash addresses.
- Satoshi Nakamoto Satoshi Nakamoto is the name used by the person or people who designed Bitcoin and created its original reference implementation.
- **SDK** Software Development Kit. A set of tools, code and documentation used by developers to create apps targeting a specific hardware or software platform.
- **Tainted Coins** Taint is a measure of correlation between two (wallet) addresses. It is only important if the user is trying to remain anonymous.
- **Testnet** *Testnet* is a network only for testing (parallel to the mainnet), test wallets, test coins, test smartnodes, test miners, and test users all simulate their mainnet counterparts in a safe environment where errors or forks are not harmful.
- Tor An anonymous routing protocol used by people wanting to hide their identity online.
- **Transaction** Some movement of data on the distributed blockchain ledger. Transactions may be divided into classical and special transactions. Similar to Bitcoin, classical transactions move balances between addresses on the blockchain.
- **Transaction Block** A collection of transactions on the SmartCash network, gathered into a block that can then be hashed and added to the blockchain.
- **Transaction Fee** A *small fee* imposed on some transactions sent across the SmartCash network. The transaction fee is awarded to the miner that successfully hashes the block containing the relevant transaction.
- **Unconfirmed Transactions** Transactions that are not yet processed by miners or held via InstantSend are "unconfirmed on the blockchain." Unconfirmed transactions can be reversed and should not be considered as "final."
- Vanity Address A SmartCash address with a desirable pattern, such as a name.

- Virgin SmartCash SmartCash received as a reward for mining a block or running a smartnodes. These have not yet been spent anywhere and are "virgin."
- Volatility The measurement of price movements over time for a traded financial asset (including SmartCash).
- **Wallet** A method of storing SmartCash for later use. A wallet holds the private keys associated with SmartCash addresses. The blockchain is the record of the SmartCash balances (and transactions) associated with those addresses.
- Whitepaper A white paper is an authoritative report or guide that informs readers concisely about a complex issue and presents the issuing body's philosophy on the matter. It is meant to help readers understand an issue, solve a problem, or make a decision.
- **Zero Confirmations** This is a transaction without any confirmations from the blockchain. It is technically reversible (unless InstantSend was used).
- vin A transaction (tx) consists of one or more inputs and one or more outputs. The vin is the list of inputs to the transaction, and vout is the list of outputs. Smartnodes require a 100000 SMART vin (exactly that amount) in order to work.

1.6 Wallets

Whenever you are storing objects with a market value, security is necessary. This applies to barter systems as well as economies using currency as a medium of exchange. While banks store balances on a private ledger, cryptocurrencies store balances under unique addresses on a distributed public ledger. The cryptographic private keys to access the balance stored on each public address are therefore the object of value in this system. This section of the documentation discusses different practical methods of keeping these keys safe in wallets, while still remaining useful for day-to-day needs.

For safety, it is not recommended to store significant funds on exchanges or software wallets. If you are holding cryptocurrency worth more than the device you use to store it, you should purchase a *hardware wallet*.

1.6.1 NodeClient Wallet

SmartCash NodeClient Wallet is the full official release of SmartCash, and supports all SmartCash features as they are released, including InstantPay and SmartNodes, as well as an RPC console and Voting features. SmartCash NodeClient Wallet (sometimes known as the QT wallet, due to the QT software framework used in development) is a professional or heavy wallet which downloads the full blockchain (several GB in size) and can operate as both a full node or smartnode on the network. Because of the requirement to hold a full copy of the blockchain, some time is required for synchronisation when starting the wallet. Once this is done, the correct balances will be displayed and the functions in the wallet can be used. SmartCash NodeClient Wallet is available for macOS, Linux, and Windows.

Features:

- InstantPay
- Wallet encryption
- Coin control and fee control
- QR code generation and address book
- SmartRewards Monitoring
- SmartNode and SmartVote
- · Automated backup
- Debug console

Available documentation:

Installation

Installing SmartCash NodeClient is as simple as going to https://smartcash.cc/wallets/#nodeclient and downloading the appropriate file for your system, then following the appropriate installation steps for your system. Detailed guides are available for Linux, macOS and Windows operating systems below.

It is also possible to *compile SmartCash NodeClient from source code*.

Linux Installation Guide

This guide describes how to download, verify, install and encrypt the SmartCash NodeClient wallet for Linux. The guide is written for Ubuntu 16.04 LTS, but the steps should be similar for other Linux distributions.

Downloading the SmartCash NodeClient wallet

Visit https://smartcash.cc/wallets/#nodeclient to download the latest SmartCash NodeClient wallet.If you are unsure whether your version of Linux is 32 or 64 bit, you can check in Ubuntu under the **System menu > About This Computer**. For details on how to check this in other versions of Linux, see here.

Once you know which version you need, download SmartCash NodeClient to your computer from https://smartcash. cc/wallets/#nodeclient.

Manually selecting and downloading SmartCash NodeClient

Save the file you downloaded to your Downloads folder.

Verifying SmartCash NodeClient

This step is optional, but recommended to verify the integrity of the file you downloaded. This is done by checking its SHA256 hash against the hash published by the SmartCash NodeClient development team.

Once the SmartCash NodeClient file, view hash in your browser and find the hash value for the SmartCash NodeClient file you downloaded.

This hash value should correspond with the hash value of the file you have downloaded to ensure it is authentic and was not corrupted during transit. To do this, open Terminal, browse to the location where you saved the file, and run the sha256sum command.



Fig. 2: Generating an SHA256 hash for the downloaded file

If the hashes match, then you have an authentic copy of SmartCash NodeClient for Linux.

View more here <https://smartcash.freshdesk.com/support/solutions/articles/35000087793>_
Extracting SmartCash NodeClient

SmartCash NodeClient for Linux is distributed as a compressed archive and not an installer. This is because this same archive also contains other files built for running a smartnode on a server, for example. In this guide, we will extract the executable file with a graphical user interface (GUI) designed for use by end users as a wallet.

Begin by creating a folder for the SmartCash NodeClient executable file on the Desktop. Browse to the Desktop (or the location of your choice) and create the folder.

Creating a folder on the Desktop Renaming the folder to SmartCash

Next, open the archive by double-clicking on it. The Archive Manager will appear. Browse to the smartcash-1.3.3/bin/ folder and extract the smartcash-qt file to the SmartCash folder you created on the Desktop by drag and drop.

To run SmartCash NodeClient for the first time, open Terminal and browse to the SmartCash folder on the Desktop, or where you chose to extract the file. Type ./smartcash-qt to run the file.

The first time the program is launched, you will be offered a choice of where you want to store your blockchain and wallet data. Choose a location with enough free space, as the blockchain can reach 10GB+ in size. It is recommended to use the default data folder if possible.

SmartCash NodeClient will then start up. This will take a little longer than usual the first time you run it, since SmartCash NodeClient needs to generate cryptographic data to secure your wallet.

Synchronizing SmartCash NodeClient to the SmartCash network

Once SmartCash NodeClient is successfully installed and started, you will see the wallet overview screen. You will notice that the wallet is "out of sync", and the status bar at the bottom of the window will show the synchronization progress.

During this process, SmartCash NodeClient will download a full copy of the SmartCash blockchain from other nodes to your device. Depending on your internet connection, this may take a long time. If you see the message "No block source available", check your internet connection. When synchronization is complete, you will see a small blue tick in the lower right corner.

You can now begin to use your wallet to send and receive funds.

Encrypting your SmartCash wallet

After your wallet has synchronized with the SmartCash network, it is strongly advised to encrypt the wallet with a password or passphrase to prevent unauthorized access. You should use a strong, new password that you have never used somewhere else. Take note of your password and store it somewhere safe or you will be locked out of your wallet and lose access to your funds.

To encrypt your wallet, click **Settings > Encrypt wallet**.

When the encryption process is complete, you will see a warning that past backups of your wallet will no longer be usable, and be asked to shut down SmartCash NodeClient. When you restart SmartCash NodeClient, you will see a small blue lock in the lower right corner.

Using the Ubuntu Repository to install SmartCash NodeClient

Ubuntu allows you to add third-party repositories to install and update software using the apt command line utility. SmartCash NodeClient team maintains such a repository, although the software version included here may be older than what is available on the website. To install SmartCash NodeClient from the repository, open the Terminal and enter the following commands:

```
sudo add-apt-repository ppa:smartcash/ppa -y && sudo apt update && sudo apt install_ {\hookrightarrow}\,{\rm smartcashd} -y
```

Raspberry Pi

install needed dependencies: ((Install of Python, a Programming Language and PIP, a package manager for Python packages))

sudo apt-get install python3-setuptools python3-pyqt5 python3-pip

Use the Python installer (pip) to download and install SmartCash Wallet:

Enter the following commands (changing the version number to match the current version as necessary)

sudo pip3 install https://github.com/SmartCash/Core-Smart/releases/download/v1.3.3/smartcash-1.3.
3-arm-rpi.tar.gz

Launch the desktop:

startx

To launch SmartCash Wallet you can either open a terminal and run:

smartcash-qt

macOS Installation Guide

This guide describes how to download, install and encrypt the SmartCash NodeClient wallet for macOS. The guide is written for macOS Sierra, but the steps should be similar for other versions.

Downloading the SmartCash NodeClient wallet

Visit https://smartcash.cc/wallets/#nodeclient to download the latest SmartCash NodeClient wallet. Save the file you downloaded to your Downloads folder.

Verifying SmartCash NodeClient

This step is optional, but recommended to verify the integrity of the file you downloaded. This is done by checking its SHA256 hash against the hash published by the SmartCash NodeClient development team. To do this, open **Terminal**, browse to the location where you saved the file, and run the following command, replacing the version with the specific version of the file you downloaded:

```
shasum -a 256 smartcash-1.3.3-osx.dmg
Generating an SHA256 hash for the downloaded file
```

If the hashes match, then you have an authentic copy of SmartCash NodeClient for macOS.

Installing SmartCash NodeClient

Open Finder and browse to your Downloads folder. Then double-click on the .dmg file you downloaded to decompress it. A window appears showing the contents of the file.

Drag the SmartCash NodeClient application file into your Applications folder to install SmartCash NodeClient.

Running SmartCash NodeClient for the first time

To run SmartCash NodeClient for the first time, either open Launchpad or browse to your **Applications** folder in Finder. Double-click **SmartCash** or **SmartCash-Qt** to start the application. You may see a warning about opening an app from an unidentified developer. To resolve this problem, simply Control-click the app icon and choose **Open** from the shortcut menu, then click **Open** again in the dialog box. The app is saved as an exception to your security settings, and you can open it in the future by double-clicking it just as you can any registered app. The first time the program is launched, you will be offered a choice of where you want to store your blockchain and wallet data. Choose a location with enough free space, as the blockchain can reach 10GB+ in size. It is recommended to use the default data folder if possible.

SmartCash NodeClient will then start up. This will take a little longer than usual the first time you run it, since SmartCash NodeClient needs to generate cryptographic data to secure your wallet.

Synchronizing SmartCash NodeClient to the SmartCash network

Once SmartCash NodeClient is successfully installed and started, you will see the wallet overview screen. You will notice that the wallet is "out of sync", and the status bar at the bottom of the window will show the synchronization progress.

During this process, SmartCash NodeClient will download a full copy of the SmartCash blockchain from other nodes to your device. Depending on your internet connection, this may take a long time. If you see the message "No block source available", check your internet connection. When synchronization is complete, you will see a small blue tick in the lower right corner.

You can now begin to use your wallet to send and receive funds.

Encrypting your SmartCash wallet

After your wallet has synchronized with the SmartCash network, it is strongly advised to encrypt the wallet with a password or passphrase to prevent unauthorized access. You should use a strong, new password that you have never used somewhere else. Take note of your password and store it somewhere safe or you will be locked out of your wallet and lose access to your funds.

To encrypt your wallet, click **Settings** > **Encrypt Wallet**. When the encryption process is complete, you will see a warning that past backups of your wallet will no longer be usable, and be asked to shut down SmartCash NodeClient. When you restart SmartCash NodeClient, you will see a small blue lock in the lower right corner.

You can now begin to use your wallet to safely send and receive funds.

Windows Installation Guide

This guide describes how to download, install and encrypt the SmartCash NodeClient wallet for Windows. The guide is written for Windows 10, but the steps should be similar for Windows XP, Vista, 7 and 8.

Downloading the SmartCash NodeClient wallet

Visit https://smartcash.cc/wallets/#nodeclient to download the latest SmartCash NodeClient wallet. You will need to manually choose your operating system and whether you need a 32 or 64 bit version. If you are unsure whether your

version of Windows is 32 or 64 bit, you can check in Windows 10 under **Start > Settings > System > About**. For details on how to check this in other versions of Windows, see here.

Under the System section, click About to view the System type. This is a 64 bit system.

Save the file you downloaded to your Downloads folder.

Verifying SmartCash NodeClient

This step is optional, but recommended to verify the integrity of the file you downloaded. To do this, open **Command Prompt**, browse to the location where you saved the file, and run the following command, replacing the version with the specific version of the file you downloaded.

certutil -hashfile smartcash-1.3.3-win64-setup.exe SHA256

If the hashes match, then you have an authentic copy of SmartCash NodeClient for Windows.

Running the SmartCash NodeClient installer

Double-click the file to start installing SmartCash NodeClient.

At this point, you may see a warning from Windows SmartScreen that the app is unrecognized. You can safely skip past this warning by clicking **More info**, then **Run anyway**.

Bypassing Windows SmartScreen to run the app. This warning is known as a "false positive".

The installer will then guide you through the installation process.

Click through the following screens. All settings can be left at their default values unless you have a specific reason to change something.

Installation is complete

Running SmartCash NodeClient for the first time

Once installation is complete, SmartCash NodeClient will start up immediately. If it does not, click **Start > SmartCash > SmartCash** to start the application. The first time the program is launched, you will be offered a choice of where you want to store your blockchain and wallet data. Choose a location with enough free space, as the blockchain can reach 10GB+ in size. It is recommended to use the default data folder if possible.

SmartCash NodeClient will then start up. This will take a little longer than usual the first time you run it, since SmartCash NodeClient needs to generate cryptographic data to secure your wallet.

Synchronizing SmartCash NodeClient to the SmartCash network

Once SmartCash NodeClient is successfully installed and started, you will see the wallet overview screen. You will notice that the wallet is "out of sync", and the status bar at the bottom of the window will show the synchronization progress. During this process, SmartCash NodeClient will download a full copy of the SmartCash blockchain from other nodes to your device. Depending on your internet connection, this may take a long time. If you see the message "No block source available", check your internet connection. When synchronization is complete, you will see a small blue tick in the lower right corner.

You can now begin to use your wallet to send and receive funds.

Encrypting your SmartCash wallet

After your wallet has synchronized with the SmartCash network, it is strongly advised to encrypt the wallet with a password or passphrase to prevent unauthorized access. You should use a strong, new password that you have never used somewhere else. Take note of your password and store it somewhere safe or you will be locked out of your wallet and lose access to your funds.

To encrypt your wallet, click **Settings > Encrypt Wallet**. When the encryption process is complete, you will see a warning that past backups of your wallet will no longer be usable, and be asked to shut down SmartCash NodeClient. When you restart SmartCash NodeClient, you will see a small blue lock in the lower right corner.

You can now begin to use your wallet to safely send and receive funds.

Video installation guides

Interface

The SmartCash NodeClient Wallet is an application that runs on your computer and allows you to make transactions on the SmartCash network. Most transactions are for sending or receiving SmartCash, but it is also possible to create signed messages or control a smartnode, for example. The SmartCash NodeClient Wallet interface is described in detail in the following sections.

The Main Window



Fig. 3: The SmartCash NodeClient Wallet

The SmartCash NodeClient window is broken up into several areas:

• The menu bar

- The tab bar
- The main area
- The status bar

The Menu Bar

The menu bar provides access to all functions of SmartCash NodeClient. There are three menus available:

File The File menu is used to manage your wallet, messages and addresses.

Settings The Settings menu provides access to wallet encryption options and general software settings.

Help The Help menu links to Debug Window, Command-line options Documents

The Tab Bar

The tab bar is used to quickly switch between the main areas of the SmartCash NodeClient. The content in the main area of SmartCash NodeClient changes depending on which tab you have selected. The following tabs are available:

The Overview tab

The overview tab offers quick access to your balance and most recent transactions.

SmartCash - Wallet	-		×
File Settings Help			
🕜 Overview 💫 Send 🖄 Receive 🕞 Transactions 🔅 SmartNodes 👼 SmartRewards 🖓 SmartVote			
Balances Recent transactions Available: 0.00000000 SMART Pending: 0.00000000 SMART Total: 0.00000000 SMART			
Syncing Headers (96.8%) 4 week(s) behind	SMAR	т 🖰 но	5 ° U

Fig. 4: The SmartCash NodeClient Overview tab

The left part of the main area is divided into two areas. The upper area shows your balances:

Available This shows your current liquid balance. This is the amount of SmartCash you can spend now.

Pending This shows funds waiting for a transaction to complete.

Immature This shows funds from smartnode or mining payments which have not yet reached the required number of confirmations.

Total This is simply your available and pending funds added together.

The right part of the screen shows your recent transactions. These are identified by icons as follows:

S Outgoing transactions



Payment to yourself



Mining or smartnode payment



Incoming transactions

- Recent incoming transactions appear black, prefixed by a + sign
- Recent outgoing transactions appear red, prefixed by a sign
- Incoming mining or smartnode payments also appear black

For more details on your transaction history, see the Transactions tab.

The Send Tab

The Send tab allows you to send funds to another address on the SmartCash network. It contains fields to enter the recipient's address, a label for the address, and the amount of SmartCash you wish to send. Options related to the transaction fee, InstantPay are also available. A quick view of your total balance is also available in the lower right corner.

The Receive Tab

The Receive tab allows you to create addresses to receive SmartCash. You can create a request for a specific amount of SmartCash or include a specific message, and send it to another user as a link or QR code.

The Transactions Tab

The transactions tab shows the entire transaction history for all addresses associated with your wallet. This appears as a table showing the time, type, label and amount of SmartCash for each transaction. You can also export the transaction history as a CSV file by clicking the Export button in the bottom right corner of the window.

The icons in the leftmost column indicate the status of the transaction. A tick indicates that the recommended number of confirmations has been passed, while a clock indicates that the transaction has yet to reach six confirmations.

8 SmartCash - Wallet	– 🗆 ×
<u>F</u> ile <u>S</u> ettings <u>H</u> elp	
🔐 Overview 💫 Send 🖄 Receive 🗇 Transactions 🔅 SmartNodes 🞯 SmartRewards 🖓 SmartVote	
Coin Control Features]
Inputs automatically selected	
Custom change address Enter a SmartCash address (e.g. SXun9XDHLdBhG4YdlueZfLfRpC9kZgwTlb)	
Pay To: Enter a SmartCash address (e.g. SXun9XDHLdBhG4YdlueZfLfRpC9kZgwTlb)	€ 🗎 🛇
Label: Enter a label for this address to add it to your address book	
A <u>m</u> ount: SMART ▼ Subtract fee from amount	
Send Send Clear All + Add Recipient InstantPay	Balance: 0.00000000 SMART
Synchronizing with network 3 week(s) behind	SMART 🔂 HD 💕 🔗

Fig. 5: The Send tab

💈 SmartCash - Wallet				-		×
<u>F</u> ile <u>S</u> ettings <u>H</u> elp						
Overview 🔊 Send	Receive Transactions	SmartNodes 🞯 Si	nartRewards 🖓 SmartVote			
Use this form to reque	st payments. All fields are optic	onal.				
Amount:	SMART V					
Reuse an existing r	receiving address (not recommer	nded) equest InstantPav				
Requested payments history	V Label	Message		Requested (SN	MART)	
					,	
Show Semove						
Synchronizing with network	3 week(s) behir	nd		SMAR	ат 🖰 но	5 0

Fig. 6: The Receive tab



Fig. 7: The transactions tab

The Status Bar

The status bar shows a synchronization progress bar and a row of status icons which indicate the status of your connection to the SmartCash network.

The Synchronization Bar

This bar shows the synchronization status of SmartCash NodeClient with the SmartCash network. Each time you open SmartCash NodeClient, it will begin downloading the blocks which have been created on the blockchain in the time since you last opened the app. These blocks are downloaded from other SmartCash users and smartnodes. If you have never opened the app before, this could mean several years' worth of blocks need downloading. The following statuses are possible:

No block source available This occurs if your internet connection is down, or if the ports required by SmartCash NodeClient are blocked by a firewall.

Synchronizing with network SmartCash NodeClient is downloading blocks from the network.

Synchronizing smartnodes database smartrewards database SmartCash NodeClient is synchronizing other data with the second layer network.

Once synchronization is complete, the progress bar will disappear and a tick will appear on the right of the status bar.

The Status Icons

۵

The lock icons indicate the status of your wallet: either locked or unlocked. You need to unlock your wallet to send funds or perform certain other actions.

ð

These icons show the synchronization status of SmartCash NodeClient with the network. Once synchronization is complete, the refresh icon will become a blue tick.

HD

These icons indicate whether your wallet is running in hierarchical deterministic (HD) mode or standard mode.

The Options Dialog

This documentation describes the functionality of the SmartCash NodeClient Options dialog, available under the **Settings > Options** menu in SmartCash NodeClient.

Main tab

The Main tab of the Options dialog contains settings related to startup and performance of the SmartCash NodeClient app.

Start SmartCash NodeClient on system login This option causes SmartCash NodeClient to start automatically when the current user logs in. On most computers there is only one main user account which is logged in automatically when the computer turns on, so this option is effectively the same as starting SmartCash Node-Client together with the operating system.

Options	?	×
Main Wallet Network Window Display		
☐ Start SmartCash on system login Size of database cache 300		
Active command-line options that override above options: none		
Reset Options OK	Cano	el

Fig. 8: The Main tab of the SmartCash NodeClient Options dialog

- **Size of database cache** This option specifies the size of the database cache in memory. A higher value will result in increased performance when adding new blocks at the cost of higher memory usage. The default value is 300MB and it should not be set lower than this level.
- Number of script verification threads This option sets the number of script verification threads, ranging from 4 to 16. Script verification is the process of following instructions recorded in the blockchain to ensure the transactions are valid. 0 means automatic and will allow script verification to scale to the number of cores available on your processor. Setting a positive number specifies that SmartCash NodeClient should use that number of processor cores, while setting a negative number will leave that number of processor cores free.

Wallet tab

The Wallet tab of the Options dialog contains settings related to how addresses are managed in the SmartCash Node-Client app. This tab allows you to configure how these addresses are used as inputs with the Coin Control.

Options	?	×
Main Wallet Network Window Display		
Expert		
Active command-line options that override above options: none		
Reset Options OK	Cance	el

- Fig. 9: The Wallet tab of the SmartCash NodeClient Options dialog
- Enable coin control features Your SmartCash NodeClient wallet balance is actually the sum total of all addresses holding balance that are associated with your wallet. When you spend SmartCash, SmartCash NodeClient will

withdraw from as many inputs as necessary to make up the desired amount of SmartCash to be transferred. This behavior may be undesirable if you want to keep a certain balance on one address. The most common use case is the requirement to maintain 100000 SmartCash on a single address as collateral for a smartnode. Enabling this option will add a button labelled **Inputs** on the **Send** tab. This provides access to the **Coin selection** dialog, which can be used to lock, unlock and prioritize different addresses in your wallet. See *here* for a more detailed explanation of Coin Control.

Spend unconfirmed change When this option is enabled, the SmartCash NodeClient wallet permits you to immediately spend change from previous transactions that has been transferred internally between addresses associated with the same wallet. This is possible even if the transaction has not yet been confirmed because the wallet knows it will eventually be confirmed since it created the internal transaction itself. Leaving this option enabled allows you to create new transactions even if previous transactions have not yet been confirmed.

Network tab

Map port using UPnP Allow incoming connections Connect through SOCKSS proxy (default proxy): Proxy IP: 127.0.0.1	Main Wallet Netw	vork <u>W</u> indow <u>D</u> isplay	
Allow incoming connections Connect through SOCKSS proxy (default proxy): Proxy IP: 127.0.0.1 Port: 9050 Used for reaching peers via: IPv4 IPv6 Tor Use separate SOCKSS proxy to reach peers via Tor hidden services: Proxy IP: 127.0.0.1 Port: 9050	Map port using UPnP		
Connect through SOCKSS proxy (default proxy): Proxy IP: 127.0.0.1 Port: 9050 Used for reaching peers via: IPv4 IPv6 Tor Use separate SOCKSS proxy to reach peers via Tor hidden services: Proxy IP: 127.0.0.1 Port: 9050	Allow incoming connect	tions	
Proxy IP: 127.0.0.1 Port: 9050 Used for reaching peers via: IPv4 IPv6 Tor Use separate SOCKSS proxy to reach peers via Tor hidden services: Proxy IP: 127.0.0.1 Port: 9050	Connect through SOC	KS5 proxy (default proxy):	
Used for reaching peers via: IPv4 IPv6 Tor Use separate SOCKSS proxy to reach peers via Tor hidden services: Proxy IP: 127.0.0.1 Port: 9050	Proxy IP: 127.0.0.1	Port: 9050	
Use separate SOCKSS proxy to reach peers via Tor hidden services: Proxy IP: 127.0.0.1 Port: 9050	Used for reaching peers vi	ia: IPv4 IPv6 Tor	
Proxy IP: 127.0.0.1 Port: 9050	Use separate SOCKS5	proxy to reach peers via Tor hidden services:	
	Proxy IP: 127.0.0.1	Port: 9050	
	Proxy IP: 127.0.0.1	Port: 9050	
	Proxy IP: 127.0.0.1	Port: 9050	
	Proxy IP: 127.0.0.1	Port: 9050	
	Proxy [P: 127.0.0.1	Port: 9050	
Active command-line ontions that override above ontions:	Proxy IP: 127.0.0.1	Port: 9050	
Active command-line options that override above options:	Proxy IP: 127.0.0.1 Active command-line option	Port: 9050	
Active command-line options that override above options: none	Proxy IP: 127.0.0.1 Active command-line option none	Port: 9050	

This tab includes options related to how your connection to the SmartCash network is made.

Fig. 10: The Network tab of the SmartCash NodeClient Options dialog

- **Map port using UPnP** This option causes SmartCash NodeClient to automatically attempt to open and map the client port on your router using UPnP (Universal Plug and Play). This feature is supported by most modern home routers and will allow you to connect to the SmartCash network without making any special settings on your router.
- Allow incoming connections This option causes your client to accept external connections. Since SmartCash is a peer-to-peer network and SmartCash NodeClient is considered a full client because it stores a copy of the blockchain on your device, enabling this option helps other clients synchronize the blockchain and network through your node.
- **Connect through SOCKS5 proxy (default proxy)** These options allow users on an intranet requiring a proxy to reach the broader internet to specify the address of their proxy server to relay requests to the internet. Contact your system administrator or check out the network settings in your web browser if you are unable to connect and suspect a proxy may be the source of the problem.
- Use separate SOCKS5 proxy to reach peers via Tor hidden services These options allow you to specify an additional proxy server designed to help you connect to peers on the Tor network. This is an advanced option for increased privacy and requires a Tor proxy on your network. For more information about Tor, see here.

Window tab

This option contains options governing behavior of the SmartCash NodeClient app window under Microsoft Windows.

Options	?	×
Main Wallet Network Window Display		
Hide tray icon Minimize to the tray instead of the tackhar		
Minimize on dose		
Active command-line options that override above options:		
none		
Reset Options QK	<u>C</u> ano	el

Fig. 11: The Window tab of the SmartCash NodeClient Options dialog

- **Hide tray icon** When this option is enabled, SmartCash NodeClient will not display an icon in the system tray. This option cannot be selected at the same time as **Minimize to the tray instead of the taskbar**.
- Minimize to the tray instead of the taskbar When this option is enabled and the SmartCash NodeClient window is minimized, it will no longer appear in your taskbar as a running task. Instead, SmartCash Core will keep running in the background and can be re-opened from the SmartCash icon in the system tray (the area next to your system clock). This option cannot be selected at the same time as **Hide tray icon**.
- Minimize on close When this option is enabled, clicking the X button in the top right corner of the window will cause SmartCash NodeClient to minimize rather than close. To completely close the app, select File > Exit.

Display tab

This tab contains options relating to the appearance of the SmartCash NodeClient app window.

Options		?	×
Main Wallet Network Window	Display		
User Interface language:	(default)		•
Unit to show amounts in:	SMART		•
Third party transaction URLs https://example.or	:om/tx/%s		
Active command-line options that override above	e options:		
none			
Reset Options	QK	<u>C</u> ano	el

Fig. 12: The Display tab of the SmartCash NodeClient Options dialog

- **User interface language** Select your preferred language from this drop-down menu. Changing the language requires you to restart the SmartCash NodeClient app.
- **Unit to show amounts in** This allows you to change the default unit of currency in SmartCash NodeClient from SmartCash to mSmartCash, µSmartCash or duffs. Each unit shifts the decimal separator three places to the right. Duffs are the smallest unit into which SmartCash may be separated.
- Third party transaction URLs This option allows you to specify and external website to inspect a particular address or transaction on the blockchain. Several blockchain explorers are available for this. To use this feature, enter the URL of your favorite blockchain explorer, replacing the %s with the transaction ID. You will then be able to access this blockchain explorer directly from SmartCash NodeClient using the context menu of any given transaction.

The Tools Dialog

This documentation describes the functionality of the SmartCash NodeClient Tools dialog, available under the **Help** ==> **Debug Window** menu in SmartCash NodeClient.

Information tab

General SmartCash Core Client name SmartCash Core Client version v1.2.6 User Agent /SmartCash Core:1.2.6/ Using BerkeleyDB version Berkeley DB 4.8.30: (April 9, 2010) Datadr C:\UsersYnoang UxpData (Roaming (Smartcash Startup time) Startup time Tue Jan 15 12:08:18 2019 Hetwork Name Number of connections 15 (n: 0 / Out: 15) Number of Smartnodes Total: 19959 (Enabled: 13935) Block chain Current number of blocks Current number of fonds 15 (3: 19: 13 2019 Hemory bool Current number of fransation Current number of fransation 2 Memory usage 5.04 KB	Information	Console	Network Traffic	Peers	Wallet Repair			
Client name SmartCash Core Client varion v1.2.6 User Agent /SmartCash Core: 1.2.6/ Using BerkeleyDB version Berkeley DB 4.8.30: (April 9, 2010) Datadir C:/Users/hoang/AppOata/Roaming/Smartcash Startup Ime Tue Jan 15 12:08:18 2019 Network main Number of connections 15 (Di: 0 / Out: 15) Number of Smartnodes Total 1999 (Enabled: 13935) Block chain Tue Jan 15 13:19:13 2019 Memory Pool Current number of fransaction 2 Memory Usage	General							
Client version v1.2.6 User Agent SmartCash Core:1.2.6/ Using BerkeleyDB version BerkeleyDB 4.8.30: (April 9, 2010) Datadir C:[Users/ixoang/AppData/Roaming/Smartcash Startup time Tue Jan 15 12:08:18 2019 Namkor main Number of connections 15 (In: 0 / Out: 15) Number of Smartodes Total: 1999 (Enabled: 13935) Block chain U Current number of foldsk 845770 Last block time Tue Jan 15 13:19:13 2019 Memory Pool Current number of sinasction 2 Memory Usage	Client name		SmartCash Co	ore				
User Agent /SmartCash Core:1.2.6/ Using BerkeleyDB version BerkeleyDB 4.8.30: (April 9, 200) Datadir C://UseryDang/ApopData Roaming/Smartcash Startup time Tue Jan 15 12:08:18 2019 Network Image: SmartDash Startup SmartDash Startup SmartDash Number of connections 15 (In: 0 / Out: 15) Number of Smartnodes Total: 19959 (Enabled: 13935) Block chain Current number of blocks Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Hemory Voal Soft KB	Client version		v1.2.6					
Using BerkeleyDB version Berkeley DB 4.8.30: (April 9, 2010) Datadr C:\Users\'hoong\AppOata\'Rooming\Smartcash Startup time Tue Jan 15 12:08:18 2019 Number of connections 15 (In: 0 / Out: 15) Number of Smartnods 15 (In: 0 / Out: 15) Bock chain Edock chain Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Memory Pool Current number of fransactions 2 Debug	User Agent		/SmartCash C	ore: 1.2.6/				
Datadir C: \Users \hoang \AppOata \Roaming \Smartcash Startup time Tue Jan 15 12:08:18 2019 Network main Number of connections 15 (D: 0 / Out: 15) Number of Smartnodes Total : 1999 (Enabled: 13935) Bock chain Bock chain Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Memory Pool Current number of transaction 2 Memory usage	Using Berkel	eyDB version	Berkeley DB 4	1.8.30: (Ap	ril 9, 2010)			
Startup time Tue Jan 15 12:08:18 2019 Network Number of connections Is (In: 0 / Out: 15) Number of Smarthodes Total: 19959 (Enabled: 13935) Block chain Current number of blocks Retwory Pool Tue Jan 15 13: 19: 13 2019 Memory Noage 5.04 KB	Datadir		C:\Users\hoa	ng\AppDat	a\Roaming\Smart	ish		
Name main Number of connections 15 (In: 0 / Out: 15) Number of Smartnodes Total: 19959 (Enabled: 19935) Block chain Current number of blocks Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Hemory Pool Current number of fransaction Current number of fransaction 2	Startup time		Tue Jan 15 1	2:08:18 20	19			
Name main Number of connections 15 (In: 0 / Out: 15) Number of Smartnodes Total: 19959 (Enabled: 13935) Block chain Enabled: 13935) Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Memory Pool Current number of transactions 2 Debug Memory usage 5.04 KB	Network							
Number of connections 15 (In: 0 / Out: 15) Number of connections Total: 19959 (Enabled: 13935) Block chain Block chain Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Memory Pool Current number of transactions 2 Debug Memory usage 5.04 KB	Name		main					
Number of Smartnodes Total: 19959 (Enabled: 13935) Block chain Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Debug Memory Voal S.04 KB Debug	Number of con	nections	15 (In: 0 / Ou	it: 15)				
Block chain Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Hemory Pool Debui Current number of transactions 2 Memory usage 5.04 KB	Number of Sm	artnodes	Total: 19959	(Enabled: 1	13935)			
Current number of blocks 845770 Last block time Tue Jan 15 13: 19: 13 2019 Memory Pool Current number of transactions 2 Memory usage 5.04 KB	Block chain							
Last block time Tue Jan 15 13:19:13 2019 Memory Pool Current number of transactions 2 Memory usage 5.04 KB	Current numbe	er of blocks	845770					
Memory Pool Debug Current number of transactions 2 Memory usage 5.04 KB	Last block time		Tue Jan 15 13	3:19:13 20	19			
Current number of transactions 2 Memory usage 5.04 KB	Memory Poo	bl						Debuc
Memory usage 5.04 KB	Current numbe	er of transact	tions 2					
	Memory usage		5.04 KB					w

Fig. 13: The Information tab of the SmartCash NodeClient Tools dialog

- **General** This section displays information on the name and version of the client and database, and the location of the current application data directory.
- Network This section displays information and statistics on the network to which you are connected.

Block chain This section shows the current status of the blockchain.

- **Memory pool** This section shows the status of the memory pool, which contains transactions that could not yet be written to a block. This includes both transactions created since the last block and transactions which could not be entered in the last block because it was full.
- **Open debug log file** This button opens debug.log from the application data directory. This file contains output from SmartCash NodeClient which may help to diagnose errors.

Console tab

The Console tab provides an interface with the SmartCash NodeClient RPC (remote procedure call) console. This is equivalent to the smartcash-cli command on headless versions of SmartCash, such as smartcashd running on a smartnode. Click the x icon to clear the console, and see the detailed documentation on RPC commands to learn about the possible commands you can issue.



Fig. 14: The Console tab of the SmartCash NodeClient Tools dialog

Network Traffic tab

The Network Traffic tab shows a graph of traffic sent and received to peers on the network over time. You can adjust the time period using the slider or **Clear** the graph.



Fig. 15: The Network Traffic tab of the SmartCash NodeClient Tools dialog

Peers tab

The Peers tab shows a list of other full nodes connected to your SmartCash Core client. The IP address, version and ping time are visible. Selecting a peer shows additional information on the data exchanged with that peer.

Informa	tion Console Netwo	ork Traffic Peers Wallet R	epair	
Vodelc	Node/Service	User Agent	Ping	
29	45.77.142.205:9678	/SmartCash Core:1.2.4/	215 ms	
30	54.38.158.249:9678	/SmartCash Core:1.2.6/	300 ms	
32	195.201.33.84:9678	/SmartCash Core:1.2.4/	208 ms	Select a peer to view detailed information.
33	139.162.166.253:9678	/SmartCash Core:1.2.4/	201 ms	
34	193.87.209.201:9678	/SmartCash Core:1.2.4/	312 ms	
35	80.211.66.238:9678	/SmartCash Core:1.2.6/	220 ms	
36	159.89.196.50:9678	/SmartCash Core:1.2.4/	55 ms	
37	209.250.255.202:9678	/SmartCash Core:1.2.6/	216 ms	
59	45.76.94.172:9678	/SmartCash Core:1.2.6/	218 ms	
63	69.166.104.36:9678	/SmartCash Core:1.2.4/	231 ms	
64	94.177.255.45:9678	/SmartCash Core:1.2.5/	233 ms	
65	107.175.102.125:9678	/SmartCash Core:1.2.5/	1114 ms	
66	199.247.5.232:9678	/SmartCash Core:1.2.6/	211 ms	

Fig. 16: The Peers tab of the SmartCash NodeClient Tools dialog

Wallet Repair tab

The Wallet Repair tab offers a range of startup commands to restore a wallet to a functional state. Selecting any of these commands will restart SmartCash NodeClient with the specified command-line option.

Tools window						-		×
Information Console Network Traffic Peers Wallet Re	epair							
Wallet repair options.								
The buttons below will restart the wallet with command-line options to transactions.	o repair the	wallet, fix iss	sues with c	orrupt block	hain files or	missing/ob	osolete	
C: \Users \hoang \AppData \Roaming \Smartcash \wallet.dat								
Salvage wallet -sa	alvagewallet	t: Attempt to	recover p	rivate keys i	from a corru	pt wallet.c	dat.	
Rescan blockchain files -re	escan: Resca	an the block	chain for m	issing walle	t transactior	IS.		
Recover transactions 1 -22	apwallettxes count owner	s=1: Recover r).	r transactio	ons from blo	ckchain <mark>(</mark> kee	:p meta-da	ata, e.g.	
Recover transactions 2 -za	apwallettxes	s=2: Recover	r transactio	ons from blo	ckchain (dro	ip meta-da	ata).	
Upgrade wallet format up	pgradewalle date of the	t: Upgrade w wallet itself!	vallet to lat)	est format o	on startup. (Note: this	is NOT ar	n
Rebuild index -re	eindex: Rebu	uild block cha	ain index fro	om current b	olk000??.dat	: files.		

Fig. 17: The Wallet Repair tab of the SmartCash NodeClient Tools dialog

Salvage wallet Salvage wallet assumes wallet.dat is corrupted and cannot be read. It makes a copy of wallet.dat to wallet.<date>.bak and scans it to attempt to recover any private keys. Check your debug.log file after running salvage wallet and look for lines beginning with "Salvage" for more information on operations completed.

- **Rescan blockchain files** Rescans the already downloaded blockchain for any transactions affecting accounts contained in the wallet. This may be necessary if you replace your wallet.dat file with a different wallet or a backup - the wallet logic will not know about these transactions, so a rescan is necessary to determine balances.
- **Recover transactions** The recover transactions commands can be used to remove unconfirmed transactions from the memory pool. Your wallet will restart and rescan the blockchain, recovering existing transactions and removing unconfirmed transactions. Transactions may become stuck in an unconfirmed state if there is a conflict in protocol versions on the network during PrivateSend mixing, for example, or if a transaction is sent with insufficient fees when blocks are full.
- **Upgrade wallet format** This command is available for very old wallets where an upgrade to the wallet version is required in addition to an update to the wallet software. You can view your current wallet version by running the getwalletinfo command in the console.
- **Rebuild index** Discards the current blockchain and chainstate indexes (the database of unspent transaction outputs) and rebuilds it from existing block files. This can be useful to recover missing or stuck balances.

Sending and receiving

Your SmartCash NodeClient Wallet is associated with a number of unique addresses that can be used to send and receive SmartCash. Each address holds its own balance, and the sum of all your balances is what appears on the **Overview** tab. When you send SmartCash, your wallet will automatically transfer funds from as many of your addresses as necessary to the destination address, which is controlled by another SmartCash user and associated with their wallet. You can control which addresses you use using the Coin Control feature.

When you confirm a transaction, SmartCash NodeClient will enter the transaction in a block, which will then be added to the blockchain for other clients to confirm. A transaction is generally considered confirmed once six blocks have been added after the block containing your transaction, although smartnode and mining payments are only released after 101 blocks. Note that a different process is used for instantpay transactions.

SmartCash addresses are 34 characters long and begin with an uppercase S.

Sending SmartCash

You can use SmartCash NodeClient to send SmartCash from your balance to another user. The receiving user will provide you with a SmartCash address to which you should send the funds. Click the **Send** tab in the tab bar and enter the destination address in the **Pay To** field.

You can also use the three icons 💷 🗐 🗭 to the right of the Pay To field to select a previously used address,
paste an address from the clipboard, or clear the current address. If this is a new address, you can enter a name for the
address in the Label field to help identify it again later. Finally, enter the amount of SmartCash you want to transfer
in the Amount field.

The other options relate to fees and instantpay. You can choose if you want to pay the network fee in addition to the amount sent, or subtract it from the amount sent. You can also increase your fee to encourage nodes on the network to prioritize your transaction. Choosing **instantpay** has a similar effect, but actually relies on a different mechanism in the second layer network to speed up the transaction time.

Let's try an example. Say you have received an invoice which you now want to pay with SmartCash. The writer of the invoice has included a SmartCash address, which can be seen in the following window beginning with *Xpa*. The invoice is for 1.45 SmartCash, which you fill in the **Amount** field.

Once you have entered the destination address and the amount, click the **Send** button. If you have encrypted your wallet, you will now be required to enter your password to unlock the wallet.

Overview Send	SmartCash - Wallet ie Settings Help	- 🗆 ×
Coin Control Features Inputs automatically selected Custom change address [Enter a SmartCash address (e.g. SXun9XDBHLdBh64Yd1ue2fLfRpC9k2gwT1b) Pay To: [Enter a SmartCash address (e.g. SXun9XDBHLdBh64Yd1ue2fLfRpC9k2gwT1b) Label: Enter a label for this address to add it to your address book movert: Subtract for from amount	🔐 Overview 💫 Send 🖄 Receive 🖃 Transactions 🛞 SmartNodes 慮 SmartRewards 🦂 SmartVote	
Inputs automatically selected Custom change address Enter a SmartCash address (e.g. SXun9XDHLdBhG4Ydlue2fLfRpC9k2gwTlb) Pay [o: Enter a SmartCash address (e.g. SXun9XDHLdBhG4Ydlue2fLfRpC9k2gwTlb) Label: Enter a label for this address to add it to your address book Label: Enter a label for this address to add it to your address book Label: Enter a label for this address to add it to your address book Label: Enter a label for this address to add it to your address book Label: Enter a label for this address to add it to your address book Label: Enter a label for this address to add it to your address book	Coin Control Features	
Custom change address Enter a SmartCash address (e.g. SXun5XDELdBh64Ydlue2fLfRpC9kZgwTlb) Pay Jo: Enter a SmartCash address (e.g. SXun5XDELdBh64Ydlue2fLfRpC9kZgwTlb) Label: Enter a label for this address to add it to your address book unwart: Subtract for from amount	Inputs automatically selected	
Pay Io: Enter a SmartCash address (e.g. SXun9XDHLdBhG4Ydlue2fLfRpC9kZgwTlb)	Custom change address Enter a SmartCash address (e.g. SXun9XDHLdBhG4YdlueZfLfRpC9kZgwTlb)	
Label: Enter a label for this address to add it to your address book	Pay Io: Enter a SmartCash address (e.g. SXun9XDRLdBhG4YdlueZfLfRpC9kZgwTlb)	
Impunt:	Label: Enter a label for this address to add it to your address book	
	Amount: SMART Subtract fee from amount	
	Send Serient InstantPay	Balance: 0.00000000 SMAR
Sgnd Olear All + Add Recipient InstantPay Balance: 0.00000000 SMAA	achronizing with network	SMART & HD

Fig. 18: The SmartCash NodeClient Send tab

SmartCash - Wallet	-		×
ile <u>S</u> ettings <u>H</u> elp			
🟠 Overview 💫 Send 🖄 Receive 🗇 Transactions 🔅 SmartNodes 🞯 SmartRewards 🖧 SmartVote			
Coin Control Features			
Inputs automatically selected			
Custom change address Enter a SmartCash address (e.g. SXun9XDHLdBhG4Ydlue2fLfRpC9kZgwTlb)			
PavTa: Enter a SmartCash address (e.g. SXun9XDHLdBhG4Ydlue2fLfRpC9kZgwTlb)	4	1	Ø
Label: Enter a label for this address to add it to your address book			
Amount: 1.45 SMART Subtract fee from amount			
Send Sclear All + Add Recipient InstantPay	Balance: 0.	00000000	SMAR
	SMA	RT 台 но	5.

Fig. 19: The Send tab filled out for a transaction

Unlock wallet	:	?	×
This operation nee	ds your wallet passphrase to unlock the wallet.		
Enter passphrase	•••••		
	OK	Can	ncel

Fig. 20: Entering the password to unlock the wallet

Finally, you are given one final confirmation and chance to cancel your send transaction before SmartCash NodeClient processes the transaction on the blockchain.

Are you sure you want to send?
1.45000000 SMART to
SRRvv2BEw1jParpMBwSHYBXsg8bmY1xMje
0.00100000 SMART added as transaction fee (0.225 kB)
Total Amount 1.45100000 SMART (=1451.00000 mSMART or 1451000.00 µSMART)

Fig. 21: Final confirmation window

If you respond with **Yes**, your transaction will be processed. Your operating system may display a notification, and the transaction will appear on the Transactions tab, where you can monitor its progress.

SMARTCASH	now
SmartCash - Sent transaction Date: 1/19/19 11:01 Amount: -0.00100000 SMART Type: Payment to yourself	

Fig. 22: The Windows 10 sent transaction confirmation notification

Note that the amount of the transaction increased by .000045 SmartCash. This is the transaction fee. In the next section, we will see what this procedure looks like from the receiving side.

Receiving SmartCash

To receive SmartCash, you must first create a receiving address to give to the sending party. To do this, click **File > Receiving addresses**. The **Receiving addresses** window appears.

				~	YU		
Use this form	to request payments.	All fields are optional					
Label:							
Amount:	•	SMART 🔻					
dessage:							
Reuse an	existing receiving add	ress (not recommended	i)				
		0					
🛱 Beg	uest payment	🗵 Clear 🔛 Requi	est InstantPay				
🛱 Beq	uest payment	Clear Requi	est InstantPay				
	uest payment	Clear Requ	est InstantPay				
Requested payment Date	ts history Label	🙁 Clear 🔄 Requ	est InstantPay	lessage		Requested (SMART)
Requested payment Date 1/15/2019 13:33	ts history Label invoice	(no message)	est InstantPay	lessage		Requested (SMART) unt requeste
Requested payment Date 1/15/2019 13:33	invoice	(no message)	est InstantPay	lessage		Requested ((no amou	SMART) unt requeste
Requested payment Date 1/15/2019 13:33	ts history Label invoice	(no message)	est InstantPay	lessage		Requested () (no amou	SMART) unt requeste
Requested payment Date 1/15/2019 13:33	ts history Label invoice	(no message)	est InstantPay	lessage		Requested ((no amou	SMART) unt requeste
Requested payment Date 1/15/2019 13:33	ts history Label invoice	(no message)	est InstantPay	lessage		Requested ((no amou	SMART) unt requeste

Fig. 23: The Receiving addresses window

Either copy an existing address by clicking on the address and then the **Copy** button, or create a new address by clicking the **New** button. You can also edit an existing address by right clicking and selecting **Edit** address from the context

menu. Give this address to the person who will send you SmartCash. Your wallet does not need to be open to receive funds, but if it is, you can watch the transaction arrive in real time. This is because your wallet constantly watches for new blocks on the blockchain when it is open, and will recognize a new transaction involving your receiving address when it occurs.





Once you have been paid, you can see the balance both on the **Overview** tab and on the **Transactions** tab.

SmartCas Sile Setting	h - Wallet							-		×
	s <u>H</u> elp Send	Receive	I	ransactions	SmartNodes	G SmartRewards	SmartVote			
All	•	All	•	Enter addres	s or label to search			Min am	ount	
Date	~	Туре		Label				Amou	nt (SMAR	F) ^
① 1/15/2	019 13:35	Received with		ზ invoice					1.000000	ou
										~
						Selected amount: (0.00000000 SMART		A Exp	port
								SMAR	ат 👌 но	5 € √

Fig. 25: The received transaction

How to Create New Receiving Addresses in SmartCashQT

Go to Receive Tab and click Request payments

instantpay

instantpay

Introduction

This documentation describes how to use instantpay to instantly send funds to any other SmartCash user around the world. SmartCash instantpay is supported by many wallets and vendors, including (but not limited to) the following:

- SmartCash NodeClient Wallet
- SmartCash Android Wallet
- SmartCash iOS Wallet
- SmartCash Web Wallet

• and many more...

You can read more about instantpay theory and processes here.

Paying with instantpay

instantpay functions by setting a flag on the transaction, causing deterministic selection of a quorum of 10 smartnodes for each input spent in an instantpay transaction. The smartnodes examine the input, and if a majority determines it has at least six confirmations, they then accept the transaction. The input is then locked until the transaction has been confirmed in six mined blocks, at which point the output can be used as an input in another instantpay transaction. This differs from inputs used in normal transactions, which can be spent after just one confirmation regardless of whether the SmartCash was received using instantpay or not. A higher fee will be charged for instantpay transactions with more than four inputs, according to the *fee schedule*. Note that the receiving wallet must also be aware of instantpay in order to be able to immediately continue with the transaction or display an appropriate notification that the transaction should be considered locked. If the receiving wallet is not aware of instantpay, it will simply appear as a normal transaction and you will need to wait for standard block confirmations.

To pay with instantpay, simply check the relevant checkbox in your app. The following screenshots indicate where this setting can be found in the SmartCash NodeClient wallets.

Is SmartCash - Wallet	- 🗆 ×
Eile Settings Help	
🏠 Overview 💫 Send 🖄 Receive 🖨 Transactions 💮 SmartNodes 👼 SmartRewards 🖧 SmartVote	
Coin Control Features	
Inputs autometically selected	
Custom change address Enter a SmartCash address (e.g. SXun9XDHLdBhG4YdlueZfLfRpC9kZgwTlb)	
PayIo: SRRvv2BEw1jParpMBwSHYEXsg8bmY1xMje	€ 🛙 🛇
Label: Enter a label for this address to add it to your address book	
Amount: 0.10000000	
Sgnd Sgnd Clear All + Add Becipient InstantPay	Balance: 1.00000000 SMAR
	SMART 🔒 HD 💕

Fig. 26: SmartCash Wallets showing instantpay option

Wallet backup and restore

Backup

This documentation describes how to safely back up your wallet file for safe storage in case your computer or laptop is damaged or lost. smartcash Core stores all data necessary to control your smartcash addresses in a single file called *wallet.dat*. This wallet is in the Berkeley DB format and stores the pairs of private/public cryptographic keys used to manage your balances on the smartcash blockchain. SmartCash NodeClient makes most of these operations transparent and even generates automatic backups of your wallet file in case it is corrupted, but the user is responsible for ensuring that these backups are stored in a safe place. **If you lose access to your wallet file, you will permanently lose access to your smartcash.**

It is important to consider that if you have not encrypted your wallet using the **Settings > Encrypt Wallet** menu item, anyone with access to the backed up wallet.dat file will immediately have full access to your smartcash. If you do

choose to encrypt your wallet, do not store the password in the same place as the wallet.dat file, particularly if you are saving the backup to the cloud.

Backup from SmartCash NodeClient

Firstly, never copy your wallet.dat file while SmartCash NodeClient is open. Always use the **File > Backup Wallet** menu if the wallet is open. When you select this menu item, a dialog box will appear to specify where the file should be saved. Enter a name for the file, select a location and click **Save**. The example below shows saving the file to a USB stick. Keep this file in a physically separate location to your computer.

<u>Settings</u> <u>H</u> elp Open <u>U</u> RI						
Open <u>U</u> RI						
		Transactions 💿 Sma	rtNodes 📾 SmartRewards 🖧 Sm			
Backup wallet		,	~ ~ ~			
Sign <u>m</u> essage			Recent transactions			
Verify message			N.			
Sending addresses			1/15/2019 13:35 invoice			
Receiving addresses						
Exit	Ctrl+Q					
		Backup Wallet				
		$\leftrightarrow \rightarrow \checkmark \uparrow$	> This PC > Documents >			~ Ū
		Organize 👻 Ne	w folder			
		A Name	^	Date modified	Туре	Size
		E Cust	om Office Templates	9/7/2018 4:09 PM	File folder	
		🕹 📃 Light	shot	9/21/2018 11:23 AM	File folder	
		Navi	cat	9/7/2018 3:18 PM	File folder	
		Uiber	Downloads	1/9/2019 4:51 PM	File folder	
		e Won	dershare hilmora	11/12/2018 5:11 PM	File folder	
		3				
		6				
		-				
		1.1				
		· ·				
		File name:	smartcash-backup-wallet			
		Course of the second	Wallet Data (8 dat)			
	4.04	ign currς	Light Caling ● Backup Wallet ● → ◆ ◆ ● Organize ▼ Ne ● ○ ∪ ↓ ● ○ ∪ ↓ ● ○ ∪ ↓ ● ○ ∪ ↓ ● ○ ∪ ↓ ● ○ ∪ ↓ ● ○ ∪ ↓	Carred Backup Wallet Backup Wallet Backup Wallet Backup Wallet Backup Wallet Drganize New folder Name Guttom Office Templates Uphthot Navicat Wondeshare Filmora File name smatcash-backup-walled File name smatcash-backup-walled	Curve Curve Curve Curve Corganize Name Date modified ViberOownloads Vib	curve Image: Starter St

Fig. 27: Backing up the SmartCash NodeClient wallet from the File menu

Backup by copying wallet.dat

If SmartCash NodeClient is not running, you can also backup your wallet by simply copying the *wallet.dat* file to another location. This file is located in the *smartcash* data folder. You were given the option to specify the location of this folder during installation, but by default the folder is in the following locations on different operating systems:

• Windows

C:\Users\YourUserName\Appdata\Roaming\smartcash

You can access this folder directly by Windows Key + R and typing %APPDATA%\smartcash

• Linux

```
/home/YourUserName/.smartcash
```

You can access this folder directly by typing cd \sim /.smartcash at the terminal or \sim /.smartcash in the path bar using the Go > Enter Location... menu item in Files

• macOS

```
/Users/YourUserName/Library/Application Support/smartcash
```

You can access this folder by typing cd $\,\sim/{\tt Library}/{\tt Application}$ Support/smartcash at the terminal or $\sim/{\tt Library}/{\tt Application}$ Support/smartcash in dialog at the Go > Go To Folder menu item in Finder

Ensure SmartCash NodeClient is not running, then simply copy the *wallet.dat* file from this folder to another folder in the normal way for your operating system. The example below shows copying the file to a USB stick using simple drag and drop while holding down **Ctrl** on a Windows system. On most operating systems, you can also right click on the file and select **Copy**, then select **Paste** in the target folder. Keep this file in a physically separate location to your computer. Be careful to copy (not move) the file!

		Search Tools	Smartcash				- 0	×	📕 I 🖸	- =	Public				
iare	View	Search						~ 👩	File	Home	Share	View			
Ton Ho	ang >	AppData > Ro	aming > Smartcash		√ Ö	Search Smartcas	sh	P	$\leftarrow \rightarrow$	~ ^	_ → One	Drive > Public			
^ N	ame	^		Date modified	Туре	Size		_	4^	Name		^	Status	Date modified	Туре
	backu	ps		1/15/2019 12:08 PM	File folder				l l					This folder is empty.	
	block	5		1/15/2019 1:07 PM	File folder										
	chain	state		1/15/2019 12:08 PM	File folder				_						
	datab	ase		12/14/2018 8:49 PM	File folder										
	rewar	ds		1/15/2019 1:17 PM	File folder										
	lock			12/14/2018 8:01 PM	LOCK File	0 KB									
1	banlis	t.dat		12/14/2018 8:02 PM	DAT File	1 KB			6						
	db			12/14/2018 8:01 PM	Text Document	0 KB			*						
	debu	,		1/15/2019 12:03 PM	Text Document	4,612 KB									
	fee_e	timates.dat		12/14/2018 8:20 PM	DAT File	17 KB									
] netful	filled.dat		12/14/2018 8:20 PM	DAT File	1 KB									
] peers	dat		1/15/2019 1:34 PM	DAT File	3,252 KB									
	smart	cash.conf		12/14/2018 8:01 PM	CONF File	1 KB		_							
	smart	node.conf		12/14/2018 8:01 PM	CONF File	T KB									
] snpay	ments.dat		12/14/2018 8:20 PM	DAT File	1 KB			Lt I						
1	versio	n.dat		12/14/2018 8:01 PM	DAT File	1 KB			•						
	wallet	.dat		1/15/2019 1:36 PM	DAT File	120 KB									
*							ſ		ii ii ii ii ii ii ii ii ii ii ii ii ii						

Fig. 28: Backing up wallet.dat by copying to another folder

Automatic backups

Every time you open SmartCash NodeClient, it will automatically create a backup copy of *wallet.dat* in the *smartcash/backups* folder. Up to 10 backups can be kept here by default, and the oldest backup will be deleted as each additional new backup is created. You can modify the number of backups kept here using the -createwalletbackups=n parameter at the command line or in *smartcash.conf*. Setting this value to 0 completely disables backups.

You can view the automatic backups folder by browsing to *smartcash* folder at the location specified above for *wallet.dat* and opening the backups folder. Since these files are not active when SmartCash NodeClient is running, you can safely copy them at any time. They are also a handy backup if the original files in the smartcash folder become corrupted due to improper shutdown of the SmartCash NodeClient app.

Restore

To restore a backup, install SmartCash NodeClient on the target system (or stop it, if already installed) and rename the existing *wallet.dat* file in the *smartcash* folder.

2	View				
on	Hoang > AppData > Roaming > Sn	nartcash	~ Ö	Search Smartcash	
	Name	Date modified	Туре	Size	
	backups	1/15/2019 12:08 PM	File folder		
	blocks	1/15/2019 1:07 PM	File folder		
	🔥 chainstate	1/15/2019 12:08 PM	File folder		
	database	12/14/2018 8:49 PM	File folder		
		1/15/2019 1:17 PM	File folder		
	lock	12/14/2018 8:01 PM	LOCK File	0 KB	
	📄 banlist.dat	12/14/2018 8:02 PM	DAT File	1 KB	
	📄 db	12/14/2018 8:01 PM	Text Document	0 KB	
	📄 debug	1/15/2019 12:03 PM	Text Document	4,612 KB	
	fee_estimates.dat	12/14/2018 8:20 PM	DAT File	17 KB	
	netfulfilled.dat	12/14/2018 8:20 PM	DAT File	1 KB	
	peers.dat	1/15/2019 1:49 PM	DAT File	3,240 KB	
	smartcash.conf	12/14/2018 8:01 PM	CONF File	1 KB	
	smartnode.conf	12/14/2018 8:01 PM	CONF File	1 KB	
	📄 snpayments.dat	12/14/2018 8:20 PM	DAT File	1 KB	
	version.dat	12/14/2018 8:01 PM	DAT File	1 KB	
	wallet.dat	1/15/2019 1:36 PM	DAT File	120 KB	

Fig. 29: Renaming the old wallet.dat file to wallet.old in the smartcash folder

Then copy the backup wallet file to the *smartcash* folder and ensure it is named *wallet.dat*. Now, when you start SmartCash NodeClient again, it will load the new wallet. Do not replace *wallet.dat* while SmartCash NodeClient is running, since this will result in data corruption!



Backup storage considerations

Any backup depends to some extent on the existence of software capable of reading the data at some future point in time. As such, it is good practice to store a copy of the software used to create the backup together with the backup file itself. In this case, this would be a copy of the version of SmartCash NodeClient you used to create the backup.

The *wallet.dat* file itself should be encrypted using a password set from the **Settings > Encrypt Wallet** menu item. However, this only prevents someone with access to the file from creating transactions, not from opening the file. You could additionally store the file in another encrypted container, such as a USB stick using BitLocker in Windows, LUKS in Linux or FileVault on macOS. It is also possible to create disk images or zip files with password encryption - the choice is yours. For further reading on encrypted containers, see here.

Where you store this file is then up to you. It could be in your home, in a safe deposit box at a bank, a waterproof or fireproof container, or on cloud storage such as Google Drive, Dropbox or iCloud. Consider where you will store any

Aame Aame Aame Aame Aame Aame Aame Aame	Date modified	Type	Scaren Sinarceasi	
ame backups blocks chainstate database rewards .lock banlist.dat db	Date modified	lvpe		
backups blocks chainstate database rewards Jock banlist.dat db		21	Size	
blocks chainstate database rewards .lock banlist.dat db	1/15/2019 12:08 PM	File folder		
chainstate database rewards j.lock bahlist.dat db	1/15/2019 1:07 PM	File folder		
database rewards Jock banlist.dat db	1/15/2019 12:08 PM	File folder		
rewards].lock] banlist.dat] db	12/14/2018 8:49 PM	File folder		
] .lock] banlist.dat] db	1/15/2019 1:17 PM	File folder		
banlist.dat db	12/14/2018 8:01 PM	LOCK File	0 KB	
db	12/14/2018 8:02 PM	DAT File	1 KB	
	12/14/2018 8:01 PM	Text Document	0 KB	
debug	1/15/2019 12:03 PM	Text Document	4,612 KB	
fee_estimates.dat	12/14/2018 8:20 PM	DAT File	17 KB	
netfulfilled.dat	12/14/2018 8:20 PM	DAT File	1 KB	
peers.dat	1/15/2019 1:49 PM	DAT File	3,240 KB	
smartcash.conf	12/14/2018 8:01 PM	CONF File	1 KB	
smartnode.conf	12/14/2018 8:01 PM	CONF File	1 KB	
snpayments.dat	12/14/2018 8:20 PM	DAT File	1 KB	
version.dat	12/14/2018 8:01 PM	DAT File	1 KB	
wallet.dat	1/15/2019 1:36 PM	DAT File	120 KB	

Fig. 30: Copying the backup file into the nodeclient folder and renaming it to wallet.dat

passwords needed to restore access to the wallet (in your head, on paper, in a password manager, etc.) and who may need access to the password in the future.

Finally it is important to understand that *wallet.dat* itself is a relatively dangerous way to store large amounts of funds - it is simply a database file storing private keys. While the convenience of storing a wallet file directly is fine for smaller holdings, it is more secure to to store large amounts of smartcash on a single predefined address in a way that you are guaranteed access through any software supporting the protocol, rather than a specific implementation of that protocol. If you are interested in this, read more about paper wallets, where the private key can be printed directly or also encrypted using BIP38 for safe storage.

Verifying backups

There is no fixed procedure to verify your backup, but you should test restoring it at least once to make sure it works. If you have a simple copy of the file, try to restore it to your current *nodeclient* folder and start SmartCash NodeClient to make sure it opens without any errors. If you decided to store the file in an encrypted zip file, make sure you can unzip it and that it opens correctly in SmartCash NodeClient. In short, make sure that you (or the person you are trusting to eventually go through this process for you) can actually reverse your backup process to get access to your smartcash, instead of relying on the fact that this process should theoretically be reversible.

Arguments and commands

All command-line options (except for -datadir and -conf) may be specified in a configuration file, and all configuration file options may also be specified on the command line. Command-line options override values set in the configuration file. The configuration file is a list of setting=value pairs, one per line, with optional comments starting with the # character.

The configuration file is not automatically created; you can create it using your favorite plain-text editor. By default, smartcash-qt (or smartcashd) will look for a file named smartcash.conf in the smartcash data directory, but both the data directory and the configuration file path may be changed using the -datadir and -conf command-line arguments.

Plat-	Path to data	Typical path to configuration file						
form	folder							
Linux	~/	/home/username/.smartcash/smartcash.conf						
ma-	~/Li-	/Users/username/Library/Application Support/smartcash/smartcash.conf						
cOS	brary/Application							
	Support/							
Win-	%APPDATA%	(Vista-10) C:\Users\username\AppData\Roaming\SmartCash\smartcash.conf						
dows		(2000-XP) C:\Documents and Settings\username\Application						
		Data\SmartCash\smartcash.conf						

Note: if running smartcash in testnet mode, the sub-folder testnet3 will be appended to the data directory automatically.

Command line arguments

These commands are accurate as of SmartCash NodeClient.

- smartcashd
- smartcash-qt
- smartcash-cli

smartcashd

SmartCash NodeClient Daemon same as

smartcash-qt

SmartCash version v1.2.6 (64-bit)

Usage:

smartcash-qt [command-line options]

Options:

-? Print this help message and exit -version Print version and exit -alertnotify=<cmd> Execute command when a relevant alert is received or we see a really long fork (%s in cmd is replaced by message) -blocknotify=<cmd> Execute command when the best block changes (%s in cmd is replaced by block hash) -checkblocks=<n> How many blocks to check at startup (default: 6, 0 = all) -checklevel=<n> How thorough the block verification of -checkblocks is (0-4, default: 3) -conf=<file> Specify configuration file (default: smartcash.conf) -datadir=<dir> Specify data directory -dbcache=<n> Set database cache size in megabytes (4 to 16384, default: 300) -loadblock=<file> Imports blocks from external blk000??.dat file on startup -maxorphantx=<n> Keep at most <n> unconnectable transactions in memory (default: 100) -maxmempool=<n> Keep the transaction memory pool below <n> megabytes (default: 300) -mempoolexpiry=<n> Do not keep transactions in the mempool longer than <n> hours (default: 72) -par=<n> Set the number of script verification threads (-4 to 16, 0 = auto, <0 = leave that many cores free, default: 0) -prune=<n> Reduce storage requirements by pruning (deleting) old blocks. This mode is incompatible with -txindex and -rescan. Warning: Reverting this setting requires re-downloading the entire blockchain. (default: 0 = disable pruning blocks, setting the setting blocks)

>1414 = target size in MiB to use for block files) –reindex–chainstate Rebuild chain state from the currently indexed blocks –reindex Rebuild chain state and block index from the blk*.dat files on disk

Connection options:

addnode= <ip></ip>	Add a node to connect to and attempt to keep the connection open
banscore= <n></n>	Threshold for disconnecting misbehaving peers (default: 100)
bantime= <n></n>	Number of seconds to keep misbehaving peers from reconnecting (default: 86400)
bind= <addr></addr>	Bind to given address and always listen on it. Use [host]:port notation for IPv6
connect= <ip></ip>	Connect only to the specified node(s)
discover	Discover own IP addresses (default: 1 when listening and no -externalip or -proxy)
dns	Allow DNS lookups for -addnode, -seednode and -connect (default: 1)
dnsseed	Query for peer addresses via DNS lookup, if low on addresses (default: 1 unless –connect)
externalip= <ip></ip>	Specify your own public address
forcednsseed	Always query for peer addresses via DNS lookup (default: 0)
listen	Accept connections from outside (default: 1 if no -proxy or -connect)
listenonion	Automatically create Tor hidden service (default: 0)
maxconnections=<	n> Maintain at most <n> connections to peers (default: 125)</n>
maxreceivebuffer=	A Maximum per-connection receive buffer, <n>*1000 bytes (default: 5000)</n>
maxsendbuffer= <n< th=""><th>> Maximum per–connection send buffer, <n>*1000 bytes (default: 1000)</n></th></n<>	> Maximum per–connection send buffer, <n>*1000 bytes (default: 1000)</n>
maxtimeadjustmer	t Maximum allowed median peer time offset adjustment. Local perspective of time may be influenced by peers forward or backward by this amount. (default: 4200 seconds)
onion= <ip:port></ip:port>	Use separate SOCKS5 proxy to reach peers via Tor hidden services (default: proxy)

-onlynet=<net> Only connect to nodes in network <net> (ipv4, ipv6 or onion) -permitbaremultisig Relay non-P2SH multisig (default: 0) -peerbloomfilters Support filtering of blocks and transaction with bloom filters (default: 1) -port=<port> Listen for connections on <port> (default: 9678 or testnet: 19678) -proxy=<ip:port> Connect through SOCKS5 proxy -proxyrandomize Randomize credentials for every proxy connection. This enables Tor stream isolation (default: 1) -rpcserialversion Sets the serialization of raw transaction or block hex returned in non-verbose mode, non-segwit(0) or segwit(1) (default: 1) -seednode=<ip> Connect to a node to retrieve peer addresses, and disconnect -timeout=<n> Specify connection timeout in milliseconds (minimum: 1, default: 5000) -torcontrol=<ip:port> Tor control port to use if onion listening enabled (default: 127.0.0.1:9051) -torpassword=<pass> Tor control port password (default: empty) –uppp Use UPnP to map the listening port (default: 0) –whitebind=<addr> Bind to given address and whitelist peers connecting to it. Use [host]:port notation for IPv6 -whitelist=<IP address or network> Whitelist peers connecting from the given IP address (e.g. 1.2.3.4) or CIDR notated network (e.g. 1.2.3.0/24). Can be specified multiple times. Whitelisted peers cannot be DoS banned and their transactions are always relayed, even if they are already in the mempool, useful e.g. for a gateway -whitelistrelay Accept relayed transactions received from whitelisted peers even when not relaying transactions (default: 1) -whitelistforcerelay Force relay of transactions from whitelisted peers even if they violate local relay policy (default: 1) -maxuploadtarget=<n> Tries to keep outbound traffic under the given target (in MiB per 24h), 0 = no limit (default: 0)

Wallet options:

-disablewallet Do not load the wallet and disable wallet RPC calls -keypool=<n> Set key pool size to <n> (default: 100) -fallbackfee=<amt> A fee rate (in SMART/kB) that will be used when fee estimation has insufficient data (default: 0.0002) -mintxfee=<amt> Fees (in SMART/kB) smaller than this are considered zero fee for transaction creation (default: 0.00001) -paytxfee=<amt> Fee (in SMART/kB) to add to transactions you send (default: 0.00) -rescan Rescan the block chain for missing wallet transactions on startup -salvagewallet Attempt to recover private keys from a corrupt wallet on startup -spendzeroconfchange Spend unconfirmed change when sending transactions (default: 1) -txconfirmtarget=<n> If paytxfee is not set, include enough fee so transactions begin confirmation on average within n blocks (default: 2) -usehd Use hierarchical deterministic key generation (HD) after BIP32. Only has effect during wallet creation/first start (default: 1) -upgradewallet Upgrade wallet to latest format on startup -wallet=<file> Specify wallet file (within data directory) (default: wallet.dat) -walletbroadcast Make the wallet broadcast transactions (default: 1) -walletnotify=<cmd> Execute command when a wallet transaction changes (%s in cmd is replaced by TxID) -zapwallettxes=<mode> Delete all wallet transactions and only recover those parts of the blockchain through -rescan on startup (1 = keep tx meta data e.g. account owner and payment request information, 2 = drop tx meta data)

Debugging/Testing options:

uacomment= <cmt></cmt>	 Append comment to the user agent string
debug= <category></category>	Output debugging information (default: 0, supplying <category> is optional). If <category> is not supplied or if <category> = 1, output all debugging information.<category> can be: addrman, alert, bench, cmpctblock, coindb, db, http, libevent, lock, mempool, mempoolrej, net, proxy, prune, rand, reindex, rpc, selectcoins, tor, zmq, qt.</category></category></category></category>
helpdebug	Show all debugging options (usage:help-help-debug)
logips	Include IP addresses in debug output (default: 0)

-logtimestamps Prepend debug output with timestamp (default: 1) -minrelaytxfee=<amt> Fees (in SMART/kB) smaller than this are considered zero fee for relaying, mining and transaction creation (default: 0.001) -max-txfee=<amt> Maximum total fees (in SMART) to use in a single wallet transaction or raw transaction; setting this too low may abort large transactions (default: 1.00) -printtoconsole Send trace/debug info to console instead of debug.log file -shrinkdebugfile Shrink debug.log file on client startup (default: 1 when no -debug)

Chain selection options:

--testnet Use the test chain

Node relay options:

-bytespersigop Equivalent bytes per sigop in transactions for relay and mining (default: 20) –datacarrier Relay and mine data carrier transactions (default: 1) –datacarriersize Maximum size of data in data carrier transactions we relay and mine (default: 223) –mempoolreplacement Enable transaction replacement in the memory pool (default: 1)

Block creation options:

--blockmaxweight=<n> Set maximum BIP141 block weight (default: 3000000)

--blockmaxsize=<n> Set maximum block size in bytes (default: 500000)

-blockprioritysize=<n> Set maximum size of high-priority/low-fee transactions in bytes (default: 50000)

RPC server options:

server	Accept command line and JSON-RPC commands
rest	Accept public REST requests (default: 0)
rpcbind= <addr></addr>	Bind to given address to listen for JSON–RPC connections. Use [host]:port no- tation for IPv6. This option can be specified multiple times (default: bind to all interfaces)
rpccookiefile= <loc></loc>	> Location of the auth cookie (default: data dir)
rpcuser= <user></user>	Username for JSON–RPC connections
rpcpassword= <pw:< th=""><th>Password for JSON–RPC connections</th></pw:<>	Password for JSON–RPC connections
rpcauth= <userpw></userpw>	• Username and hashed password for JSON–RPC connections. The field <userpw> comes in the format: <username>:<salt>\$<hash>. A canoni- cal python script is included in share/rpcuser. This option can be specified multi- ple times</hash></salt></username></userpw>
rpcport= <port></port>	Listen for JSON–RPC connections on <port> (default: 9679 or testnet: 19679)</port>
rpcallowip= <ip></ip>	Allow JSON–RPC connections from specified source. Valid for $\langle ip \rangle$ are a single IP (e.g. 1.2.3.4), a network/netmask (e.g. 1.2.3.4/255.255.255.0) or a network/CIDR (e.g. 1.2.3.4/24). This option can be specified multiple times
rpcthreads= <n></n>	Set the number of threads to service RPC calls (default: 4)

UI Options:

-choosedatadir Choose data directory on startup (default: 0) -lang=<lang> Set language, for example "de_DE" (de-fault: system locale) -min Start minimized -rootcertificates=<file> Set SSL root certificates for payment request (default: -system-) -splash Show splash screen on startup (default: 1) -resetguisettings Reset all settings changed in the GUI

smartcash-cli

- == Addressindex == getaddressbalance getaddressdeltas getaddressmempool getaddresstxids getaddressutxos
- **== Blockchain ==** getbestblockhash getblock "hash" (verbose) getblockchaininfo getblockcount getblockhash index getblockhashes timestamp getblockheader "hash" (verbose) getblockheaders "hash" (count verbose) getchaintips getdifficulty getmempoolinfo getrawmempool (verbose) getspentinfo gettxout "txid" n (includemempool) gettxoutproof ["txid",...] (blockhash) gettxoutsetinfo verifychain (checklevel numblocks) verifytxoutproof "proof"
- == Control == debug (011addrmanlalertlbenchlcoindbldbllocklrandlrpclselectcoinslmempoollmempoolrejlnetlproxylprunelhttpllibeventl) getinfo help ("command") stop
- == Generating == generate numblocks (maxtries) getgenerate setgenerate generate (genproclimit)
- == Mining == getblocktemplate (SigningAddress) getmininginfo getnetworkhashps (blocks height) prioritisetransaction <txid> <priority delta> <fee delta> submitblock "hexdata" ("jsonparametersobject")
- == Network == addnode "node" "addlremovelonetry" clearbanned disconnectnode "node" getaddednodeinfo dummy ("node") getconnectioncount getnettotals getnetworkinfo getpeerinfo listbanned ping setban "ip(/netmask)" "addlremove" (bantime) (absolute) setnetworkactive truelfalse

- == Rawtransactions == createrawtransaction [{"txid":"id","vout":n},...] {"address":amount,"data":"hex",...} (locktime) decoderawtransaction "hexstring" decodescript "hex" fundrawtransaction "hexstring" (options) getrawtransaction "txid" (verbose) sendrawtransaction "hexstring" (allowhighfees instantsend) signrawtransaction "hexstring" ([{"txid":"id","vout":n,"scriptPubKey":"hex","redeemScript":"hex"},...] ["privatekey1",...] sighashtype)
- == Smartcash == smartmining "command"... smartnode "command"... smartnodebroadcast "command"... smartnodelist ("mode" "filter") smartrewards "command"... snsync [statuslnextlreset] spork <name> [<value>]
- == Util == createmultisig nrequired ["key",...] estimatefee nblocks estimatepriority nblocks estimatesmartfee nblocks estimatesmartpriority nblocks validateaddress "smartcashaddress" verifymessage "smartcashaddress" "signature" "message"
- == Wallet == abandontransaction "txid" addmultisigaddress nrequired ["key",...] ("account") backupwallet "destination" dumphdinfo dumpprivkey "smartcashaddress" dumpwallet "filename" getaccount "smartcashaddress" getaccountaddress "account" getaddressesbyaccount "account" getbalance ("account" minconf include-Watchonly) getnewaddress ("account") getrawchangeaddress getreceivedbyaccount "account" (minconf) getreceivedbyaddress "smartcashaddress" (minconf) gettransaction "txid" (includeWatchonly) getunconfirmedbalance getwalletinfo importaddress "address" ("label" rescan p2sh) importelectrumwallet "filename" index importprivkey "smartcashprivkey" ("label" rescan) importpubkey "pubkey" ("label" rescan) importwallet "filename" instantsendtoaddress "smartcashaddress" amount ("comment" "comment-to" subtractfeefromamount) keypoolrefill (newsize) listaccounts (minconf includeWatchonly) listaddressgroupings listlockunspent listreceivedbyaccount (minconf includeempty includeWatchonly) listreceivedbyaddress (minconf includeempty includeWatchonly) listsinceblock ("blockhash" target-confirmations includeWatchonly) listtransactions ("account" count from includeWatchonly) listunspent (minconf maxconf ["address",...]) lockunspent unlock ([{"txid":"txid","vout":n},...]) move "fromaccount" "toaccount" amount (minconf "comment") sendfrom "fromaccount" "tosmartcashaddress" amount (minconf "comment" "comment-to") sendmany "fromaccount" {"address":amount,... } (minconf "comment" ["address",...]) sendtoaddress "smartcashaddress" amount ("comment" "comment-to" subtractfeefromamount) setaccount "smartcashaddress" "account" settxfee amount signmessage "smartcashaddress" "message" walletlock walletpassphrase "passphrase" timeout walletpassphrasechange "oldpassphrase" "newpassphrase"

Advanced topics

Coin Control

Coin Control allows users of the SmartCash NodeClient Wallet to specify which addresses and Unspent Transaction Outputs (UTXOs) should be used as inputs in transactions. This allows you to keep a specific balance on certain addresses in your wallet, while spending others freely. In smartcash NodeClient Wallet, click **Settings > Options > Wallet > Enable coin control features**. Now, when you go to the Send tab in your wallet, a new button labelled **Inputs...** will appear. Click this button to select which UTXOs can be used as input for any transactions you create. The following window appears:

Right click on the transaction(s) you do not want to spend, then select **Lock unspent**. A small lock will appear next to the transaction. You can click the **Toggle lock state** button to invert the locked/unlocked state of all UTXOs. When you are ready to continue, click **OK**. You can now safely create transactions with your remaining funds without affecting the locked UTXOs.

Coin Selection Quantity: 1 Amount: L00000000 SMART Fee: ≈0.00022600 SMART After Fee: ≈0.99 Bytes: ≈226 Priority: low Dust: no Change: ≈0.89 (un)selectal O Tree mode Ist mode Ist mode Amount Received with label Received with address Date Confirmations 1.00000000 invoice SP3WsLqWkmeyWikLfVjzBo7p8o4wWFZr1A 1/15/2019 13:35 13	7 1977400 SMART 1977400 SMART Priority Iow
Amount Amount L0000000 SMART Fee: x0.0002600 SMART After Fee: x0.99 bytes: x226 Priority: bw Dust: no Change: x0.89 (un)select al O Tree mode Iostmode Iostmode Iostmode Iostmode Iostmode Amount Received with label Received with address Date Confirmations I 1.00000000 invoice SP3WsLqWkmeyWikLfrigtBo7p8o4wWFZr1A 1/15/2019 13:35 13	977400 SMART 1977400 SMART Priority Iow
Vrtes: v225 Priority: low Dust: no Change: =0.89 (un)select all O: Tree mode: Io List mode Io List mode <td< td=""><td>P77400 SMART Priority Iow</td></td<>	P77400 SMART Priority Iow
(m)select all O Tree mode Ist mode Amount Received with label Received with address Date Confirmations Interview 1,0000000 invoice SP3WsLqWkmeyWikLYlgB07p804wWFZr1A 1/15/2019 13:35 13	Priority Iow
Amount Received with label Received with address Date Confirmations I.00000000 invoice SP3WsLqWkmeyWikLfVgB07p8o4wWFZr1A 1/15/2019 13:35 13	Priority low
Amount Received with label Received with address Date Confirmations I.00000000 invoice SP3WsLqWkmeyWikLfVjzBo7p804wWFZr1A 1/15/2019 13:35 13	Priority Iow
Amount Received with label Received with address Date Confirmations Image: Confirmation of the second of	Priority low
☑ 1.00000000 invoice SP3Ws1qWkmeyWikLYjgBo7p8o4wWF2r1A 1/15/2019 13:35 13	low

Fig. 31: Coin Selection window in SmartCash NodeClient wallet, showing two smartnodes (testnet)

Specific Amount I about Specific Amount Specific Amount Specific Specific O mount Specific Specific Specific Specific Specific	Specify 1 About 1000000 3001 Pare 1000000 3001 Operating 1 Operating 2000000000000000000000000000000000000	Constraint Processe Processe Processe Processe Processe Image: Constraint Processe Processe Processe Processe Image: Constrat Processe Processe <td< th=""><th>Security Amount Limitation provid Face Limitation provid Address : Limitation provid Security Foreignet Oversetting Security Descrity Descrity</th><th>Specify 1 Available 1000000 point Norw 1000000 Available 1000000 point Orace 1000000000000000000000000000000000000</th><th> Competen </th><th>on</th><th></th><th></th><th></th><th></th><th></th><th></th><th>7 ×</th></td<>	Security Amount Limitation provid Face Limitation provid Address : Limitation provid Security Foreignet Oversetting Security Descrity Descrity	Specify 1 Available 1000000 point Norw 1000000 Available 1000000 point Orace 1000000000000000000000000000000000000	 Competen 	on							7 ×
New T New F New F Output INIT Lipitanizi Oran wei Statistic Statistic Lipitanizi Statistic Statistic Statistic Lipitanizi Statistic Statistic Statistic	Agent 1 Topology area Topology area Topology area Carpy area Carpy area Strain of Carpy area Strain of Carpy area Carpy area Carpy area Strain of Carpy area Strain of Carpy area Carpy area Strain of Carpy area Strain of Carpy area Strain of Carpy area Carpy area Strain of Carpy area Strain of Carpy area Strain of Carpy area	Name: Name: Data to Da	Note: Note: Note: Section: Note: Note:	Note 0 Date	Quantity: 0		Amount: 0.0000	000 SHART	Fee: 0.00000000 SHART		After Fee: 0.0	0000000 SMART	
Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>	Contraction Date Bit standard Image: Second and the standard Second and the standard Date Definition Point of the standard	Constant France & Barrow Intervant France and Water Conferential States Conferential States Intervant Symbol conferential States Conferential States Date Conjustic Symbol conference Symbol conference Symbol conference Conjustic Symbol conference Symbol conference Symbol conference Experiment Conference Symbol conference Symbol conference	Composition Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>	Egents O travel: @ strate Image: Constant Strate Descent Strate	Bytes: 0		Priority: none		Dust: no		Change: 0.0	0000000 SMART	
Associal Decard Activity Date Continuence Party 1000000000000000000000000000000000000	Jacquer Found - Mound	Jagement Posed with the sound with t	Jagorov Product all Multiple Data all multiple Data Jagorov 99064cpthong0kliftyda/blok/0564 V15201138 1 Copy shared Copy shared 1 Copy shared Copy shared 1 Copy shared Image: Copy shared 1 Copy shared Image: Copy shared 1	Fund Fund with an and a final state in the second state	(un)select al	O Tree mode 🛞	Ustmode						
			<	✓ ✓ ∞ ● Can Sitellan ✓ Quartity: 0 Annual: 0.000000 59487 Fee: 10		Amount Re Loop and Copy and Copy and Copy and Copy tax Lock ang Unlock or	ceived with label	Facelved with address SP3mLqWareyWitlPy	zBoTpBoHwWFZr1A	Dubs 1/15/2019 12:35	Confirmation 13	es Pischy Iow	
 	Constitution Constitution Quantity: 0 Answer: 8.300000.90477 Fee: 8 Price: 0 Private, nove Data	Con Statistics Quarking: 8 Annualt: 5 000000 54427 Fore 10. Bytes: 0 Priority: nove Dust: no	Bytes 0 Priority none Diust n		OK.								
Constitutions Constitutions Constitutions Quantity: 0 Meanity: 0 Prende: 0 Private Defence: 0 Defence: 0 Defence: 0	Con Selection Quartyr: 0 Annuel: 0.000000 59407 Fire: 0.1 Prets 0 Provide United Detter [polymerical One mode (Lobard)	Consistention Quantity: 0 Perform 0	bytes 0 proverty rore but n [srbatecial] ○ Tree mode (black	(surfacebect tail O Tree mode (@ List mode (1 locked)	OK.								
	Can Settler Can Settler Generative Development Succession SHAT Free SA Free D Priority: rove Development (and Can Settler Priority D Priority: rove Development Annount Received with Address	Concidentian Quantity: 0 Annuet: 0.000000 5947 Fee: 0 Prefat: 0 Prefat: 0 Prefat: 0 Concidential Office roots @ Listmade (Libded) [Indext] Office root @ Listmade (Libded) [Indext] Received with Address	Protect 0 Product n Protect n Product n Protect n	parbalectar 2 ⊂ Titre mode ⊂ Bi Lat.mode ⊂ (1 bidad) Amount Received with bade Received with address	<u> </u>								

Fig. 32: Locking UTXOs in SmartCash NodeClient wallet

HD Wallets

Since version v1.2.3, SmartCash NodeClient has included an implementation of BIP39/BIP44 compatible hierarchical deterministic (HD) key generation. This functionality is only available from the command line by specifying the usehd option when starting SmartCash NodeClient for the first time. Use this function with care, since the mnemonic seed and keys will be stored in plain text until you specify a wallet passphrase. Note that the wallet passphrase is different to the mnemonic passphrase, which is often also referred to as the "25th word" or "extension word". The wallet passphrase encrypts the wallet file itself, while the mnemonic passphrase is used to specify different derivation branches from the same mnemonic seed.

We will use the Windows GUI wallet in this example, but the commands are similar if using smartcash-qt or smartcashd on other operating systems. Enter the following command to get started with a randomly generated HD wallet seed and no mnemonic passphrase:

smartcash-qt.exe --usehd=1

A new HD wallet will be generated and SmartCash NodeClient will display a warning informing you that you must encrypt your wallet after verifying it works correctly. Open the console from **Help -> Debug Windows - >Consolse tab** or issue the following RPC command from smartcash-cli to view the mnemonic seed:

dumphdinfo

SmartCash NodeClient will display the HD seed in both hexadecimal and as a BIP39 mnemonic. To restore an existing HD wallet, or define your own separately generated mnemonic and/or passphrase, ensure no wallet.dat file exists in the datadir and enter the following command:

The HD wallet will be restored and your balance will appear once sync is complete.

Multiple wallets

It is possible to select between different smartcash wallets when starting SmartCash NodeClient by specifying the wallet argument, or even run multiple instances of SmartCash NodeClient simultaneously by specifying separate data directories using the datadir argument.

To begin, install the SmartCash NodeClient wallet for your system according to the *installation instructions*. When you get to the step **Running SmartCash NodeClient for the first time**, you can decide whether you want to maintain separate wallet.dat files in the default location (simpler if you do not need to run the wallets simultaneously), or specify entirely separate data directories such as e.g. C:\smartcash1 (simpler if you do want to run the wallets simultaneously).

Separate wallet.dat files

For this scenario, we will create two shortcuts on the desktop, each using a different wallet file. Navigate to the binary file used to start SmartCash NodeClient (typically locatd at C:\Program Files\smartcash\smartcash-qt.exe or similar) and create two shortcuts on the desktop. Then open the **Properties** window for each of these shortcuts.



Fig. 33: Creating desktop shortcuts using Windows 10

Modify the **Target** property of each shortcut to point to a different wallet file by specifying the wallet argument when starting the wallet. If you do not specify a wallet argument, wallet.dat will be used by default. The specified wallet file will be created if it does not exist. The following example demonstrates two wallets named workwallet.dat and homewallet.dat:

• Wallet Target 1: -wallet=workwallet.dat "C:\Program Files\smartcash\smartcash-qt.exe"

• Wallet Target 2: -wallet=homewallet.dat "C:\Program Files\smartcash\smartcash-qt.exe"

			٤.	2	
👂 smartcash1 Pro	operties	×	🕫 smartcash2 Pr	operties	×
Security General	Details Shortcut tcash1	Previous Versions Compatibility	Security General	Details Shortcut tcash2	Previous Versions Compatibility
Target type: A Target location: S Target: 4 Start in: 5 Shortcut key: 1 Run: 1 Comment: 0 Open File Loc	pplication martCash urtCash\amatcash.qt.e (C:\Program Files\Sma tone Vormal window ation Change loc	xe wallet-workwallet.dat"	Target type: // Target location: 5 Target: [Stat in: [Shotcut key: [Flun: [Comment: [Open Fife Loc	ppilcation imetCash rtCash\umatCash\qt.et 'C:\Program Files\Sma Nome Nome Nome ation Change for	xe wallet-homewallet dat * ftCash**
	ОК	Cancel Apply		ОК	Cancel Apply

Fig. 34: Specifying separate wallet files

You can now use the two icons to quickly and easily open different wallets from your desktop. Note that you cannot open both wallets simultaneously. To do this, you will need two separate data directories, as described below.

Separate data directories

Start SmartCash NodeClient and allow it to synchronize with the network, then close SmartCash NodeClient again. You can now create two directories at e.g. C:\smartcash1 and C:\smartcash2 and copy the blocks and chainstate directories from the synchronized data directory into the new directories. Each of these will serve as a separate data directory, allowing you to run two instances of SmartCash NodeClient simultaneously. Create two (or more) shortcuts on your desktop as described above, then specify arguments for datadir as shown below:

- Datadir Target 1: "C:\Program Files\nodeclient\smartcash-qt.exe" -datadir=C:\smartcash1 -listen=0
- Datadir Target 2: "C:\Program Files\nodeclient\smartcash-qt.exe" -datadir=C:\smartcash2 -listen=0

You can now use the two icons to quickly and easily open different wallets simultaneously from your desktop. Both wallets maintain separate and full copies of the blockchain, which may use a lot of drive space. For more efficient use of drive space, consider using an SPV or "light" wallet such as *smartcash Electrum* to maintain multiple separate wallets without keeping a full copy of the blockchain.

Secure

Securing your SmartCash QT wallet is as important as securing the physical wallet. Same principles apply but with SmartCash QT wallet more attack vectors are present. Because of it, it is more important to pay attention to securing your funds against attackers and malicious software.

Security	Details	Previous Versions	Security	Details	Previous Versions
General	Shortcut	Compatibility	General	Shortcut	Compatibility
😰 sn	nartcash1		smart	cash2	
Target type:	Application		Target type: A	oplication	
Target location	: SmartCash		Target location: Si	nartCash	
Target:	nartcash-qt.exe -datad	ir=C:\smartcash1 -listen=0"	Target: 1	artcash-qt.exe -datad	ir=C:\smartcash2 -listen=
Start in:	"C:\Program Files\Sma	rtCash"	Start in:	C:\Program Files\Sma	rtCash"
Shortcut key:	None		Shortcut key:	lone	
Run:	Normal window	~	Run: N	lomal window	
Comment:			Comment:		
Open File L	ocation Change Ic	on Advanced	Open File Loca	tion Change Ic	on Advanced

Fig. 35: Specifying separate datadirs



Fig. 36: Two instances of SmartCash NodeClient running simultaneously

Password Protection

Every single SmartCash QT wallet should be encrypted with a strong password! Encrypting the wallet means that the data file holding private keys (such as SmartCash QT's wallet.dat) is not readable by default. When the file gets stolen the attacker can't easily extract keys and spend your SmartCash.

He'd need to brute force the file which takes a lot of time - thus the importance of a strong password. It should be reasonably long (at least 16 chars) and comprises of numbers, lowercase, uppercase and special characters.

Click https://www.howtogeek.com/195430/how-to-create-a-strong-password-and-remember-it/ here and here https://www.cloudwards.net/how-to-set-up-a-strong-password/ for more info on strong passwords.

Store smaller Amounts

As with your physical wallet, you shouldn't have all eggs in one basket. It's a good practice to load your regular mobile/computer wallet with smaller amounts and replenish it when needed. The rest should be kept somewhere secure in a cold storage. (SmartCash are working with some cold storage provider to support SmartCash)

Keep it Up to Date

Every software has bugs and SmartCash QT wallets are no exception. Make sure to check for updates regularly as they bring new features and most importantly security and stability fixes. You should use modern operating systems such as Linux, Mac OS X, iOS and Android to keep your software updated automatically.

Backup

Always backup your wallet and store it in a safe place. *nodeclient-backup* It prevents you from losing your SmartCash in a case of a hardware and software failure. It also lets you recover funds after your PC or mobile was stolen or lost.

SmartCash - Wallet Eile Setting Halp	-		×
Coverview 🌮 Send 🖄 Receive 🖃 Transactions 💮 SmartNodes 🙉 SmartRewards 🖧 SmartVote			
Balances Recent transactions Available: 1.00000000 SMART b 1/15/2019 13:35 Pending: 0.00000000 SMART b 1/15/2019 13:35 Total: 1.00000000 SMART b 1/15/2019 13:35	+1.000	00000 SM	IART
	SMAF	т 🛆 но	5 ∢

Fig. 37: NodeClient Wallet
1.6.2 Electrum Wallet

SmartCash Electrum is a light wallet which uses powerful external servers to index the blockchain, while still securing the keys on your personal computer. Transactions are verified on the SmartCash blockchain using a technique called Secure Payment Verification (SPV), which only requires the block headers and not the full block. This means that wallet startup is almost instant, while still keeping your funds secure and mobile. It does not currently support advanced InstantPay features.

SmartCash Electrum is a fork of the Electrum wallet for Bitcoin. While this documentation focuses on using Smart-Cash Electrum, full documentation of all Bitcoin Electrum features (mostly identical in SmartCash Electrum) is available at the official documentation site.

Installation

Download

You can download SmartCash Electrum from the official SmartCash website.

- https://smartcash.cc/wallets/#electrum
- https://github.com/SmartCash/electrum-smart/releases

You can optionally verify the integrity of your download by running the following commands (example for Linux):

Linux

SmartCash Electrum for Linux is available from a PPA for Ubuntu and Linux Mint, and as a source tarball for other systems. As of version 4.0, it requires Python 3 to run. Enter the following commands to install from PPA:

```
sudo add-apt-repository ppa:smartcash/electrum-smart
sudo apt update
sudo apt install electrum-smart
```

Enter the following commands (changing the version number to match the current version as necessary) in the terminal to install SmartCash Electrum from the source tarball:

macOS

Simply download and run the DMG file. You may need to grant permission to install, depending on your security settings. Click through the installation wizard and run SmartCash Electrum from your Applications folder when complete.

Windows

Simply download and run the installer file to set up SmartCash Electrum. You may need to grant permission to install, depending on your security settings. Click through the installation wizard and run SmartCash Electrum from the Start menu when complete.

Creating a New Wallet

SmartCash Electrum gathers configuration data when run for the first time. For more on the concepts behind this process, skip to the later sections of this guide discussing backups, security, and addresses. When setting up SmartCash Electrum for the first time, a wizard will guide you through the process of creating your first wallet. The first screen asks how you would like to connect to the remote server. Select **Auto connect** and click **Next** to continue. You will see a notice that no wallet currently exists. Enter a name for your wallet (or accept the default name) and click **Next** to create your wallet.

Electrum-	Smart - Wizard	?	\times
	Create wallet_1 What kind of wallet do you want to create? Standard wallet Monort SmartCash addresses or private keys		
	Cancel	Nex	(t

Fig. 38: Selecting the server and naming your first wallet

You will be asked what kind of wallet you want to create. Choose between **Standard wallet**, **Multi-signature wallet** and **Import SmartCash addresses or private keys**. If you are unsure, select **Standard wallet** and click **Next** to continue. You will then be asked how you want to store/recover the seed. If stored safely, a seed can be used to restore a lost wallet on another computer. Choose between **Create a new seed**, **I already have a seed**, or **Use a hardware device**. If you are using Electrum SmartCash for the first time and not restoring an existing wallet, choose **Create a new seed** and click **Next** to continue.

mart - Wizard			?	×			
Create wallet_3							
What kind of wallet do you w	ant to create?						
Standard wallet							
Multi-signature wallet							
 Import SmartCash addres 	sses or private keys						
		Canad	Novet				
		Curren	Next				
Electrum-	Smart - Wizard	Cuncer	IVEAU		_	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new • Create a new seed ○ I already have a seed	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new © Create a new seed O I already have a seed O Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new © Create a new seed O I already have a seed O Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new Create a new seed I already have a seed Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new Create a new seed I already have a seed Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new Create a new seed I already have a seed Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new Create a new seed I already have a seed Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	×
Electrum-	Smart - Wizard Keystore Do you want to create a new Create a new seed I already have a seed Use a hardware device	w seed, or to restore a	wallet usin	g an ex	isting seed?	?	

Fig. 39: Selecting the wallet type and keystore

Electrum SmartCash will generate your wallet and display the recovery seed. Write this seed down, ideally on paper and not in an electronic format, and store it somewhere safe. This seed is the only way you can recover your wallet if you lose access for any reason. To make sure you have properly saved your seed, Electrum SmartCash will ask you to type it in as a confirmation. Type the words in the correct order and click **Next** to continue.

¢

Electrum-	Smart - Wizard		?	×	
	Your wallet generation seed is:	april country believe pen six wolf e	state		
	Please save these 12 words on paper (order is in your wallet in case of computer failure.	nportant). This seed will allow you t	Options to recover		
	WARNING: • Never disclose your seed. • Never type it on a website. • Do not store it electronically.				
		Back	Nex	t	
	Electrum-Smart - Wizard	I			?
	Your seed is that you have	nportant! If you lose your seed, y e properly saved your seed, please	our money w retype it he	ill be perm re.	anently lost. To make sure
	S	adapt dad daily			

Fig. 40: Generating and confirming the recovery seed

A password optionally secures your wallet against unauthorized access. Adding a memorable, strong password now improves the security of your wallet by encrypting your seed from the beginning. Skipping encryption at this point by not selecting a password risks potential theft of funds later, however unlikely the threat may be. Enter and confirm a password, ensure the Encrypt wallet file checkbox is ticked and click Next to continue.

Your SmartCash Electrum wallet is now set up and ready for use.

Sending and receiving

You may own SmartCash stored in another software wallet, or on an exchange such as exmo or hitbtc, or simply want to send or receive funds as a wage or business transaction. Funds can be transferred between these source and the Electrum wallet using SmartCash addresses. Your wallet contains multiple addresses, and will generate new addresses as necessary. Since the SmartCash blockchain is transparent to the public, it is considered best practice to use a new address for each transaction in order to maintain your privacy.

Back

Next

Х



Fig. 41: Entering and confirming a wallet encryption password

<u>W</u> allet	View To	ois <u>H</u> elp								
History	🚿 Send	Receive	SmartNodes	🖧 SmartVote	SmartRewards	Addresses	🤳 Coins	@	Contacts	

Fig. 42: SmartCash Electrum after setup is complete

Sending

Click the **Send** tab to make a payment. Enter the destination address in the **Pay to** field, either manually or by pasting from the clipboard. Optionally enter a **Description** for to appear in your transaction history, followed by the **Amount** to be sent. The total amount of the transaction is the sum of the sent amount and transaction fee, which is calculated automatically. SmartCash Electrum issues a warning if the total transaction amount exceeds the wallet balance.

Fig. 43: Transaction ready to send in SmartCash Electrum wallet

The wallet will request your password, then broadcast the transaction to the network and display a confirmation dialog with your transaction ID.

Enter Password ? × Amount to be sent: 0.5 SMART Mining fire: DOUI SMART Enter your password to proceed Password Cancel OK			
Amount to be sent: 0.5 SMART Mining fee: 0.001 SMART Password [🔮 Enter Password 🛛 🗙		
Enter your password to proceed Password Cancel OK Pasyment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb	Amount to be sent: 0.5 SMART Mining fee: 0.001 SMART		
Password Cancel OK Second OK Payment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bt OK	Enter your password to proceed		
Cancel OK	Password		
 Information ? Payment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb 	Cancel OK		
Information ? Payment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb OK			
Payment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb OK		🕸 Information	?
Payment sent. 17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb OK			
17fcb832c2fff4b486019ffcc9c87e335e669db900442245eed0751208bb		Payment sent.	
OK		17fcb832c2fff4b486019ffcc9c87e335e669db90044224	45eed0751208bl
OK			
			OK

Fig. 44: Password prompt and transaction confirmation in SmartCash Electrum wallet

Receiving

You can view your receiving addresses by clicking the **Receive** tab. Double-click the **Receiving address**, then copy it to the clipboard by clicking the Copy to clipboard icon. If you intend to use the address repeatedly, you can also enter a description click Save to store the address in the Requests list. Clicking an address in the list will display the stored information in the top area, together with a QR code containing the same information.

🔅 El	ectrum-SMART 4.0).0 - default_wall	et [standard]						- 0	×
<u>F</u> ile	Wallet View	ools <u>H</u> elp								
N 🔤	istory 🛛 🚿 Send	Receive	SmartNodes	🖧 SmartVote	SmartRewards	Addresses	J Coins	Contacts	Console	
^	Date	Description	on				A	mount	Balance	
25	Unconfirmed [444.	4 sat/b]						-0.501	4.49	9
•	2019-01-15 14:54							+5.	5.	
¥	2019-01-15 14:32							-1.982	0.	
¥	2019-01-15 13:35							-1.001	1.98	2
۷.	2018-11-26 11:02	test ma	ilsmart					-0.004	2.98	3

Fig. 45: Transaction ready to send in SmartCash Electrum wallet

Enter this address in the software sending the funds, send it to the person transferring funds to you or scan it directly from your mobile wallet. Once the transaction is complete, the balance will appear in the upper left corner of your wallet, and the indicator in the **History** table will change from **Pending** to **Paid**.



Fig. 46: Successfully received payment in SmartCash Electrum wallet

Once you have used an address, you can either continue using it or click New to generate a new address.

Monitoring transactions

The **History** tab lists all current and pending transactions. A transaction to an address in your wallet will appear in the list soon after it is made. Initially, this transaction will be marked as **Unconfirmed**, followed by a clock indicator on the left. As the SmartCash network processes the transaction, the status will update in the transaction history list. The network confirms transactions with a new block roughly every 5 seconds, and a transaction is considered confirmed (and therefore spendable) after six confirmations. These processed transactions are denoted with a green checkmark and the timestamp at which the transaction was made.



To view additional transaction details, right click a transaction on the **History** tab and select **Details** from the context menu. You can also use this menu to copy the transaction ID to the clipboard (this can be used as proof that a given transaction occurred), edit the transaction description for your records or view the transaction on an external block explorer.



Fig. 47: SmartCash Electrum wallet History tab immediately after receiving a transaction and after confirmation is complete

II Transaction	?	×
Transaction ID:		
19f9b05caeca039bf4fcfc1724a7d3760f519f96d4b8480a4d403a346150f1ef		
Status: 15 confirmations		
Date: 2019-01-15 14:32		
Amount sent: 1.981 SMART		
Size: 339 bytes		
Fee: 0.001 SMART (294. sat/byte)		
LockTime: 845822		
Inputs (2)		
bfod48917b08ed61:1 SNr2YVXXy2288J37KdR3XsGjvg2uT1Rbp2c de9c4eb57a0ddaa9:1 SQ93LdNu3g2mXgvehpxMYAw4F4S616YLAg		
Outputs (1)]
SRRvv2BEwljParpMBwSHYBXsg8bmY1xMje 1.981		
Copy 🔛 Export Sign Broadcast Save	C	lose

Fig. 48: Transaction details in SmartCash Electrum wallet

Wallet security

Change password

To change the wallet's password, select the **Wallet > Password** option from the main menu, or click on the lock icon in the lower right of the main window. Enter and confirm a new secure password. Should you forget your wallets' password, all is not lost. Your wallet can be restored in its entirety using the backup procedure described here.

Backup

In SmartCash Electrum, a seed is a complete backup of all addresses and transactions. Access your wallet's seed through the seed icon in the lower right of the main screen, or the **Wallet > Seed** main menu option. When prompted, enter the secure password you chose when setting up the SmartCash Electrum wallet.

l Info	mation	ive		SmartNodes	SmartVote	SmartRewards	Addresses	Je Coins	Contacts	Console	
Pace	word							A	mount	Balance	
Eass	word								-0.501	4.499	
Deice	ta ha a								+5.	5.	
Priva	te keys	·							-1.982	0.	
Add	esses	+							-1.001	1.982	
Labe	ls	▶ sma	irt						-0.004	2.983	
Histo	ory	🖡 vn	twitte	en -					-1.001	2.987	
Con	acts	, 11e	t						-1.001	3.988	
Invo	ces								-0.001	4.989	

Fig. 49: Displaying the wallet recovery seed in SmartCash Electrum

Hand-copy the twelve words found in the box to a piece of paper and store it in a safe location. Remember, anyone who finds your seed can spend all of the funds in your wallet.



Fig. 50: Viewing the recovery seed

Alternatively, a backup file can be saved using the **File > Save Copy** main menu option. This file stores the wallet's encrypted seed along with any imported addresses. Restoring this backup will require the wallet password.

Restore

The only thing needed to recover a SmartCash Electrum wallet on another computer is its seed. You can test wallet recovery with your current installation of SmartCash Electrum by selecting the **File > New/Restore** menu item. A dialog will appear asking you to name your new wallet. Enter a name, select **Standard wallet** as the wallet type and then choose **I already have a seed**.



Fig. 51: Restoring a wallet from an existing seed

Next, copy the twelve word seed into the text field.



Fig. 52: Entering the recovery seed

If your seed was entered correctly, SmartCash Electrum gives you the option to add a password for your wallet. After restoring your wallet, SmartCash Electrum will list any existing transactions from this wallet. This process may take a few minutes, and the transactions may appear as **Not Verified**. This problem disappears after restarting the program.

To restore a wallet file without using the recovery seed, copy the file to the application data folder according to your operating system:

- Linux: Open Files, select Go > Go to folder, copy the path ~/.electrum-smart and paste it into the dialog box.
- macOS: Open Finder, select Go > Go to Folder, copy the path ~/.electrum-smart and paste it into the dialog box.
- Windows: Open Explorer, copy the path %APPDATA% \Electrum-Smart and paste it in to the address bar.

Frequently Asked Questions

How does SmartCash Electrum work?

SmartCash Electrum focuses on speed, low resource usage and providing a simple user experience for SmartCash. Startup times are instant because it operates in conjunction with high-performance servers that handle the most complicated parts of the SmartCash system.

Does SmartCash Electrum trust servers?

Not really; the SmartCash Electrum client never sends private keys to the servers. In addition, it verifies the information reported by servers using a technique called Simple Payment Verification.

What is the Seed?

The seed is a random phrase that is used to generate your private keys. Example:

constant forest adore false green weave stop guy fur freeze giggle clock

Your wallet can be entirely recovered from its seed. To do this, select the I already have a seed option during startup.

How secure is the seed?

The seed created by SmartCash Electrum has 128 bits of entropy. This means that it provides the same level of security as a SmartCash private key (of length 256 bits). Indeed, an elliptic curve key of length n provides n/2 bits of security.

What are change addresses?

The SmartCash Electrum wallet design and workflow are based on a concept called a "wallet generation seed". This seed is a unique, randomly- selected list of twelve words. A SmartCash Electrum wallet uses its seed as a template for generating addresses.

To understand the problem that seeds solve, browse to the Electrum **Receive** tab. Next, open the collapsible entry marked **Change**.

Notice that the total balance does not only show the sum of all receiving addresses, but also the separately listed **Change** addresses. Where did these new change addresses come from and why does the first one now hold funds?

Hist	orv 🖌 Send 🚣 Peceive	(ii) SmartNodes	SmartVote	R SmartDewards	Addresses	- Coine	Contacte	Console	
er: All		· · · · · · · · · · · · · · · · · · ·	ye and trote	C) Sharacenards		Cons		CONSOL	×
ype	Address	Lab	el				Balance		Tx
ceiving	SimqhPqM7xkjrjExYFoMsRU	njam41J5ahx					0.		2
ceiving	SdyTJU4Mn1nLrY3yPCfe62t	RZy2fsPexfx					0.		2
eiving	Sbgq1f1hZvtU6RZVixVhc4E	R61KEkey5t5					0.		2
ceiving	SXDtJs2H9hrPxAsxqBpn1s1	zfBhHbapXMK					0.		0
eiving	SeaJ2xxFA8hddwntRvh3r3h	boVmJyTTvxg					0.		0
eiving	SQ33N8Jx7cGbgGRbSkySyhX	G11TwbSeKYC					0.		0
eiving	SdVrP5iazchjuzof5NA84SQ	PFKXcC4cSht					0.		0
eiving	STAp6wXVFNMEA7WodScCr3n	qBX9pz5HdLm					0.		0
eiving	SULt3bDpTM5SVfG4hyPwQPD	UzGmwjtfy3V					0.		0
eiving	SYpzaocRY1Qr2XyG89ir1v8	z PuMaCGPPQS					0.		0
eiving	SaR5j26UD1D43TjRdemhy4C	eKQPfn63H1T					0.		0
eiving	STcGjgUdxMvNjPv7HNSiqXQ	YSH5skcARS8					0.		0
eiving	SSez 4mkDFwUn9rmuQK7ZEHg	GU7tvuZsmo3					0.		0
ange	SPpGeY7UeGoGy68XLLVD4f2	hzQynuUWCTX					0.		2
ange	ShWudX2PG1wpj3T5aHS1VRV	kcdMZe2pFrT					0.		2
ange	SVd9VzaJ5oFz8ekT3DtjbAS	bkn8zwa18yi					0.		2
ange	SNr2YvXxyzZe8LJ7KdRJxGj	vgZuT1Rbp2c					0.		2
ange	SQ93LaNu3g2mXgvehpxMYAw	4F4S616YLAg					0.		2
ange	SV4yMZam3TK5CQdF6LKLp9e	200LCVW1nTq					4.	499	1
nge	SVJZa21nSqGQqAyKGQ7Vrm3	xwRy71TTYQd					0.		0
ange	STY4yx6nFxG7vLZaCSPJ8r1	ZIYFNCDWGXX					0.		0
ange	SgsiFESD2/SXOAc1sagMpb9	wiripaniCdm					0.		0

Fig. 53: Receiving and change addresses in SmartCash Electrum

SmartCash is an electronic cash system, meaning that it shares much in common with the process of using paper banknotes. Although some cash payments involve exact change, many do not. You tend to "overpay" when using cash, and expect to receive the difference as change. Perhaps surprisingly, this is how SmartCash transactions work as well. If the entire balance of an address is not required for any given transaction, the remainder is sent to a new and unused address under control of the same wallet. This address is generated deterministically (rather than randomly) from the wallet seed, which means that any other wallet will also regenerate the change addresses in the same order from the same recovery seed, and have access to the balances.

Spending the entire balance and sending any remainder to a change address is considered good practice because it prevents the transaction recipient from linking transactions by browsing the blockchain, thus compromising your privacy. If privacy is not a concern, change addresses can be disabled via the **Tools > preferences** menu option.

How can I send the maximum available in my wallet?

Type an exclamation mark (!) in the **Amount** field or simply click the **Max** button. The fee will be automatically adjusted for that amount.

How can I send SmartCash without paying a transaction fee?

You can create a zero fee transaction in the GUI by following these steps:

- Enable the Edit fees manually option
- Enter 0 in the Fee field
- Enter the amount in the **Amount** field

Note that transactions without fees might not be relayed by the SmartCash Electrum server, or by the SmartCash network.

Is there a way to enter amounts in USD in SmartCash Electrum?

Yes, go to **Tools > Prefernce > Fiat** and select a **Fiat currency** to display the current exchange rate from the chosen **Source**.

4	Preferences			?	×
	Transactions	Appearance	Fiat	Identity	
	Fiat currency		USD		•
	Show history rai	tes			
	Show capital gai	ins in history			
	Show Fiat balan	ce for addresses			
	Source		CoinGe	ecko	•
				Close	2

Fig. 54: Entering values in fiat currency in SmartCash Electrum

What does it mean to "Freeze" an address in SmartCash Electrum?

When you freeze an address, the funds in that address will not be used for sending SmartCash. You cannot send SmartCash if you don't have enough funds in your non-frozen addresses.

How is the wallet encrypted?

SmartCash Electrum uses two separate levels of encryption:

- Your seed and private keys are encrypted using AES-256-CBC. The private keys are decrypted only briefly, when you need to sign a transaction; for this you need to enter your password. This is done in order to minimize the amount of time during which sensitive information is unencrypted in your computer's memory.
- In addition, your wallet file may be encrypted on disk. Note that the wallet information will remain unencrypted in the memory of your computer for the duration of your session. If a wallet is encrypted, then its password will be required in order to open it. Note that the password will not be kept in memory; SmartCash Electrum does not need it in order to save the wallet on disk, because it uses asymmetric encryption (ECIES).

Wallet file encryption is activated by default since version 2.8. It is intended to protect your privacy, but also to prevent you from requesting SmartCash on a wallet that you do not control.

I have forgotten my password but still have my seed. Is there any way I can recover my password?

It is not possible to recover your password. However, you can restore your wallet from its seed phrase and choose a new password. If you lose both your password and your seed, there is no way to recover your money. This is why we ask you to save your seed phrase on paper.

To restore your wallet from its seed phrase, create a new wallet, select the type, choose **I already have a seed** and proceed to input your seed phrase.

Can I import private keys from other SmartCash clients?

In SmartCash Electrum 4.0, you cannot import private keys in a wallet that has a seed. You should sweep them instead.

If you want to import private keys and not sweep them you need to create a special wallet that does not have a seed. For this, create a new wallet, select **Use public or private keys**, and instead of typing your seed, type a list of private keys, or a list of addresses if you want to create a watching-only wallet. A master public (xpub) or private (xprv) will also work to import a hierarchical deterministic series of keys. You will need to back up this wallet, because it cannot be recovered from seed.

Electrum-	Smart - Wizard	?	×
	Import SmartCash Addresses Enter a list of SmartCash addresses (this will create a watching-only wallet), or a list of g keys. dfgdfgdfsgsdgsfgsdgdsfg	private	
	Back	Next	

Fig. 55: Importing a list of private keys to create a wallet

Can I sweep private keys from other SmartCash clients?

Sweeping private keys means to send all the SmartCash they control to an existing address in your wallet. The private keys you sweep do not become a part of your wallet. Instead, all the SmartCash they control are sent to an address that has been deterministically generated from your wallet seed.

To sweep private keys go to **Wallet > Private Keys > Sweep**. Enter the private keys in the appropriate field. Leave the **Address** field unchanged. This is the destination address from your existing SmartCash Electrum wallet. Click on **Sweep**. SmartCash Electrum then takes you to the **Send** tab where you can set an appropriate fee and then click on **Send** to send the coins to your wallet.

Where is my wallet file located?

The default wallet file is called default_wallet and is created when you first run the application. It is located under the */wallets* folder.

- Linux: Open Files, select Go > Go to folder, copy the path ~/.electrum-smart and paste it into the dialog box
- macOS: Open Finder, select Go > Go to Folder, copy the path ~/.electrum-smart and paste it into the dialog box
- Windows: Open Explorer, copy the path %APPDATA%\Electrum-SMART and paste it in to the address bar

Can I do bulk payments with SmartCash Electrum?

You can create a transaction with several outputs. In the GUI, type each address and amount on a line, separated by a comma.

Ø Electrum-SMA	ART 4.0.0 - default_wal	let [standard]						- 0	×
<u>F</u> ile <u>W</u> allet <u>V</u> ie	ew <u>T</u> ools <u>H</u> elp								
🖳 History 🚿	Send 🕹 Receive	SmartNodes	SmartVote	SmartRewards	Addresses	🔳 Coins	Contacts	Console	
Pay to									1 1 1 1
Description									
Amount		SMART		USD	Max	¢			
Fee		SMART	Pay to many Enter a list One outpu Format a You may li	7 of outputs in the 'Pay it per line. differs, amount Dad a CSV file using th	v X toʻfield. e file icon.	Send			
3alance: 4.499 SMAR	T (0.06 USD) 1 SMART~	0.01 USD					ľ	3 🏀	

Fig. 56: Creating a transaction with multiple outputs in SmartCash Electrum

Amounts are in the current unit set in the client. The total is shown in the GUI. You can also import a CSV file in the **Pay to** field by clicking on the folder icon.

Can SmartCash Electrum create and sign raw transactions?

SmartCash Electrum lets you create and sign raw transactions right from the user interface using a form.

SmartCash Electrum freezes when I try to send SmartCash

This might happen if you are trying to spend a large number of transactions outputs (for example, if you have collected hundreds of donations from a SmartCash faucet). When you send SmartCash, SmartCash Electrum looks for unspent coins that are in your wallet in order to create the new transaction. Unspent coins can have different values, much like physical coins and bills.

If this happens, you should consolidate your transaction inputs by sending smaller amounts of SmartCash to one of your wallet addresses; this would be the equivalent of exchanging a stack of nickels for a dollar bill.

What is the gap limit?

The gap limit is the maximum number of consecutive unused addresses in your deterministic sequence of addresses. SmartCash Electrum uses it in order to stop looking for addresses. In SmartCash Electrum 4.0, it is set to 20 by default, so the client will get all addresses until 20 unused addresses are found.

How can I pre-generate new addresses?

SmartCash Electrum will generate new addresses as you use them, until it hits the gap limit.

If you need to pre-generate more addresses, you can do so by typing wallet.create_new_address() in the console. This command will generate one new address. Note that the address will be shown with a red background in the address tab, to indicate that it is beyond the gap limit. The red color will remain until the gap is filled.

WARNING: Addresses beyond the gap limit will not automatically be recovered from seed. To recover them will require either increasing the client's gap limit or generating new addresses until the used addresses are found.

If you wish to generate more than one address, you may use a 'for' loop. For example, if you wanted to generate 50 addresses, you could do this:

```
for x in range(0, 50):
    print wallet.create_new_address()
```

How to upgrade SmartCash Electrum?

Warning: always save your wallet seed on paper before doing an upgrade.

To upgrade SmartCash Electrum, just *install* the most recent version. The way to do this will depend on your OS. Note that your wallet files are stored separately from the software, so you can safely remove the old version of the software if your OS does not do it for you.

Some SmartCash Electrum upgrades will modify the format of your wallet files. For this reason, it is not recommended to downgrade SmartCash Electrum to an older version once you have opened your wallet file with the new version. The older version will not always be able to read the new wallet file.

Advanced functions

SmartCash Electrum is based on Electrum, a Bitcoin wallet. Most functions are identical, which means it is not necessary to reproduce the entirety of the Electrum documentation here. The following sections describe some frequently used advanced functions. For further details on other advanced functions in Electrum for both Bitcoin and SmartCash, please click the links below.

- Electrum documentation
- Electrum seed version system
- Electrum protocol specification
- Serialization of unsigned or partially signed transactions
- Simple Payment Verification
- The Python Console
- Using Electrum Through Tor

Smartnodes in SmartCash Electrum

SmartCash Electrum supports smartnode creation through an interface called the Smartnode Manager.

Smartnode Manager

The Smartnode Manager can be accessed via Smartnodes tab. This manager displays the status of your smartnode(s). A wallet with no Smartnodes will begin with a default smartnode for which you can fill in the necessary information.

The manager displays the following data about each smartnode you have set up:

- The alias (name) of the smartnode.
- The Address.
- The protocol version that your smartnode supports.
- The status of the smartnode (e.g. whether it has been activated).
- The Payee.

Smartnode setup

A smartnode requires a "smartnode" key, which is known to both SmartCash Electrum and your smartnode. Your smartnode will use this key to sign messages, and the SmartCash network will know that you authorized it to. A smartnode key can either be one of your SmartCash Electrum keys, or an imported key.

IP address and Alias

Certain information about your smartnode is required. The IP address and port that your smartnode uses must be supplied. Also, the Alias that your smartnode. This information is filled in automatically if you import a "smartnode.conf" file.

🖪 Create new	Smartnode	?	×
Alias	MyNode1		
IP-Address	000.000.000		
Select a colla	teral for your new node		
Address TX	-Index TX-Hash		
Smartnode	ey /Rzdk/vceFmekFZ1fE79UaATtjhZm/NDbki86F3PAYGrK85YjhFn Copy SmartnodeKey Custom SmartnodeKey		
Its required to prompted.	use the "Smartnode Key" above when you install your new node. You can manually insert it into your node's smartcash.conf or provide it to the bash ins	taller wh	en
Cancel	Apply		

Fig. 57: Entering IP and Alias information

Collateral

To start a smartnode, you must have a 100000 SMART payment available in your wallet. You can scan your wallet for 100000 SMART payments in the **Create Smartnode** tab of the Smartnode Manager.

After scanning, a list of available 100000 SMART collateral payments will be displayed. Selecting one of them will cause the selected smartnode's data to be filled in, though these changes won't be saved until you click the **Save** button in the lower-right corner of the tab.

Create n	ew Smartnode		?	×
Alias	MyNode 1			
IP-Addres	s 000.000.000			
Select a o	ollateral for your new node			
Address	TX-Index	TX-Hash		
Smartnod	e Kev 7RbzshaT26GFYvhxBRKzK1nLrft6KHmbtSwtTvp8JCvZZ5GSGFf	Copy SmartnodeKey Custom SmartnodeKey		
Its required prompted.	to use the "Smartnode Key" above when you install your new node. Yo	can manually insert it into your node's smartcash.conf or provide it to the b	ash installer w	vhen
Cancel	Apply			

Fig. 58: Entering IP and alias

Activating your smartnode

After selecting a collateral payment and specifying a delegate key, you can activate your smartnode. Do this by clicking **Activate Smartnode** in the **Activate Smartnode** tab of the Smartnode Manager. If the **Activate Smartnode** button cannot be clicked, look at the message in the **Status** bar. It will show you why your smartnode cannot be activated.

Activation will require your password if your wallet is encrypted, because a message must be signed. After waiting for SmartCash Electrum to sign and broadcast your smartnode announcement, you will be presented with a message detailing the result. The status of your smartnode will be updated in the table and the **View Smartnode** tab.

Create new	Smartnode	?	×
Alias	MyNode 1		
IP-Address	000.000.000.000		
Select a coll	ateral for your new node		
Address T	K-Index TX-Hash		
Smartnode	Key 7RbzshqT26GFYyhx8RKzK1nLrff6KHmb1SwtTvp83CyZZ5GSGFf Copy SmartnodeKey Custom SmartnodeKey		
Its required to prompted.	use the 'Smartnode Key' above when you install your new node. You can manually insert it into your node's smartcash.conf or provide it to the bash ins	taller wh	ien
Cancel	Apply		

Fig. 59: Entering IP and protocol information

Importing smartnode.conf

You can import a *smartnode.conf* file using the **Smartnode.conf** tab of the Smartnode Manager. This is the recommended way of setting up Smartnodes, as it allows you to configure Smartnodes for SmartCash Core and SmartCash Electrum in the same way. Importing a *smartnode.conf* file will automatically set up one or more smartnode configurations in the Smartnode Manager.

Multisig wallets

This tutorial shows how to create a 2 of 2 multisig wallet. A 2 of 2 multisig consists of 2 separate wallets (usually on separate machines and potentially controlled by separate people) that have to be used in conjunction in order to access the funds. Both wallets have the same set of addresses.

- A common use-case for this is if you want to collaboratively control funds: maybe you and your friend run a company together and certain funds should only be spendable if you both agree.
- Another one is security: one of the wallets can be on your main machine, while the other one is on a offline machine. That way you make it very hard for an attacker or malware to steal your coins.

Create a pair of 2-of-2 wallets

Each cosigner needs to do this: In the menu select **File > New**, then select **Multi-signature wallet**. On the next screen, select 2 of 2.

Electrum-S	Smart - Wizard	?	×
	Multi-Signature Wallet Choose the number of signatures needed to unlock funds in your wallet: From 2 cosigners Require 2 signatures		
	Back	Nex	t

Fig. 60: Selecting x of y signatures for a multi-signature wallet

After generating and confirming your recovery seed, you will be shown the xpub address for this wallet.

After generating a seed (keep it safely!) you will need to provide the master public key of the other wallet. Of course when you create the other wallet, you put the master public key of the first wallet.

You will need to do this in parallel for the two wallets. Note that you can press cancel during this step, and reopen the file later.

Receiving

Check that both wallets generate the same set of Addresses. You can now send to these **Addresses** (note they start with a "7") with any wallet that can send to P2SH Addresses.



Fig. 61: xpub key of the first wallet

Electrum-	Smart - Wizard	?	\times
	Add Cosigner 2 Please enter the master public key (xpub) of your cosigner. Enter their master private lif you want to be able to sign for them.	key (xprv)
	sdfasdfasfasdfasfasfsdf		
	Back	Nex	

Fig. 62: Entering xpub from the second wallet in the first wallet

Sending

To spend coins from a 2-of-2 wallet, two cosigners need to sign a transaction collaboratively. To accomplish this, create a transaction using one of the wallets (by filling out the form on the **Send** tab). After signing, a window is shown with the transaction details.

The transaction now has to be sent to the second wallet. Several options are available for this:

• You can transfer the file on a USB stick

You can save the partially signed transaction to a file (using the **Save** button), transfer that to the machine where the second wallet is running (via USB stick, for example) and load it there (using **Tools > Load transaction > From file**)

• You can use QR codes

A button showing a QR code icon is also available. Clicking this button will display a QR code containing the transaction, which can be scanned into the second wallet (**Tools > Load Transaction > From QR Code**)

With both of the above methods, you can now add the second signature to the transaction (using the **Sign** button). It will then be broadcast to the network.

Sweep a paper wallet

You may have received a paper wallet as a gift from another SmartCash user, or previously stored one in a safe deposit box. Funds are swept from a *paper wallet* into a live wallet by importing its private key, which is a long sequence of characters starting with the number "7" or the capital letter "X". The example below displays a private key (WIF format).



Fig. 63: Public address and associated private key produced by SmartCash Paper Wallet Generator

Funds from paper wallets are swept into an SmartCash Electrum Wallet by creating a transaction using the private key and sending it to a new address from your wallet. This is necessary because it is not possible to add new public or private keys to an existing deterministic series of addresses derived from a seed phrase.

Begin by selecting the **Wallet > Private Keys > Sweep** menu item. The **Sweep private keys** dialog will appear, where you can paste your private key(s). An unused address controlled by your SmartCash Electrum wallet appears in the lower field, and can be changed by clicking the **Address** button. Once you have pasted your private key, click the **Sweep** button.

SmartCash Electrum then prepares a transaction using the private key you just imported to derive the public address for the transaction input and the address from your wallet as the output, and signs the message. Click **Broadcast** to enter the transaction on the blockchain. The balance will then appear in your wallet under the specified address. The address you swept is left with zero balance.

🚳 Sweep private keys	?	×
Enter private keys:		
VKDG86VvhBw4KHcXXuCUkCntZu8636V8Sp31dHajTMGzCwygaVry		
		iii 💼
Address Sbgq1f1hZvtU6RZVixVhc4ER61KEkey5t5		
Cancel	Swe	ep

Fig. 64: Entering the private key

Command line

SmartCash Electrum has a powerful command line available when running under Linux or macOS. This section will show you a few basic principles.

Using the inline help

To see the list of SmartCash Electrum commands, type:

```
electrum help
```

To see the documentation for a command, type:

```
electrum help <command>
```

Magic words

The arguments passed to commands may be one of the following magic words: ! ? : -.

The exclamation mark *!* is a shortcut that means 'the maximum amount available'. Note that the transaction fee will be computed and deducted from the amount. Example:

electrum payto SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5 !

A question mark ? means that you want the parameter to be prompted. Example:

electrum signmessage SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5 ?

Use a colon : if you want the prompted parameter to be hidden (not echoed in your terminal). Note that you will be prompted twice in this example, first for the private key, then for your wallet password:

electrum importprivkey :

A parameter replaced by a smartcash - will be read from standard input (in a pipe):

cat LICENCE | electrum signmessage Sbgq1f1hZvtU6RZVixVhc4ER61KEkey5t5 -

Aliases

You can use DNS aliases in place of bitcoin addresses, in most commands:

```
electrum payto ecdsa.net !
```

Formatting outputs using jq

Command outputs are either simple strings or json structured data. A very useful utility is the 'jq' program. Install it with:

```
sudo apt-get install jq
```

The following examples use it.

Sign and verify message

We may use a variable to store the signature, and verify it:

```
sig=$(cat LICENCE| electrum signmessage SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5 -)
```

And:

```
cat LICENCE | electrum verifymessage SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5 $sig -
```

Show the values of your unspents

The *listunspent* command returns a list of dict objects, with various fields. Suppose we want to extract the *value* field of each record. This can be achieved with the jq command:

electrum listunspent | jq 'map(.value)'

Select only incoming transactions from history

Incoming transactions have a positive 'value' field:

electrum history | jq '.[] | select(.value>0)'

Filter transactions by date

The following command selects transactions that were timestamped after a given date:

```
after=$(date -d '07/01/2015' +"%s")
electrum history | jq --arg after $after '.[] | select(.timestamp>($after|tonumber))'
```

Similarly, we may export transactions for a given time period:

Encrypt and decrypt messages

First we need the public key of a wallet address:

```
pk=$(electrum getpubkeys Sbgq1f1hZvtU6RZVixVhc4ER61KEkey5t5| jq -r '.[0]')
```

Encrypt:

cat | electrum encrypt \$pk -

Decrypt:

```
electrum decrypt $pk ?
```

Note: this command will prompt for the encrypted message, then for the wallet password.

Export private keys and sweep coins

The following command will export the private keys of all wallet addresses that hold some SmartCash:

electrum listaddresses --funded | electrum getprivatekeys -

This will return a list of lists of private keys. In most cases, you want to get a simple list. This can be done by adding a jq filer, as follows:

electrum listaddresses --funded | electrum getprivatekeys - | jq 'map(.[0])'

Finally, let us use this list of private keys as input to the sweep command:

```
electrum listaddresses --funded | electrum getprivatekeys - | jq 'map(.[0])' |_

→electrum sweep - [destination address]
```

Using cold storage with the command line

This section will show you how to sign a transaction with an offline SmartCash Electrum wallet using the command line.

Create an unsigned transaction

With your online (watching-only) wallet, create an unsigned transaction:

electrum payto SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5 0.1 --unsigned > unsigned.txn

The unsigned transaction is stored in a file named 'unsigned.txn'. Note that the –unsigned option is not needed if you use a watching-only wallet.

You may view it using:

```
cat unsigned.txn | electrum deserialize -
```

Sign the transaction

The serialization format of SmartCash Electrum contains the master public key needed and key derivation used by the offline wallet to sign the transaction. Thus we only need to pass the serialized transaction to the offline wallet:

cat unsigned.txn | electrum signtransaction - > signed.txn

The command will ask for your password, and save the signed transaction in 'signed.txn'.

Broadcast the transaction

Send your transaction to the SmartCash network, using broadcast:

cat signed.txn | electrum broadcast -

If successful, the command will return the ID of the transaction.

How to accept SmartCash on a website using SmartCash Electrum

This tutorial will show you how to accept smartcash on a website with SSL signed payment requests. It is updated for SmartCash Electrum 2.6.

Requirements

- A webserver serving static HTML
- A SSL certificate (signed by a CA)
- Electrum version >= 2.6

Create a wallet

Create a wallet on your web server:

electrum create

You can also use a watching only wallet (restored from xpub), if you want to keep private keys off the server. Once your wallet is created, start SmartCash Electrum as a daemon:

electrum daemon start

Add your SSL certificate to your configuration

You should have a private key and a public certificate for your domain. Create a file that contains only the private key:

```
-----BEGIN PRIVATE KEY-----
your private key
-----BEGIN END KEY-----
```

Set the path to your the private key file with setconfig:

```
electrum setconfig ssl_privkey /path/to/ssl.key
```

Create another file that contains your certificate and the list of certificates it depends on, up to the root CA. Your certificate must be at the top of the list, and the root CA at the end:

```
-----BEGIN CERTIFICATE-----
your cert
-----END CERTIFICATE-----
intermediate cert
-----BEGIN CERTIFICATE-----
-----BEGIN CERTIFICATE-----
root cert
-----END CERTIFICATE-----
```

Set the *ssl_chain* path with setconfig:

```
electrum setconfig ssl_chain /path/to/ssl.chain
```

Configure a requests directory

This directory must be served by your webserver (eg Apache):

electrum setconfig requests_dir /var/www/r/

By default, SmartCash Electrum will display local URLs, starting with 'file://' In order to display public URLs, we need to set another configuration variable, *url_rewrite*. For example:

electrum setconfig url_rewrite "['file:///var/www/','https://electrum.org/']"

Create a signed payment request

```
electrum addrequest 3.14 -m "this is a test"
{
    "URI": "smartcash:SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5?amount=3.14&r=https://
    electrum.org/r/7c2666541a",
    "address": "SbgqlflhZvtU6RZVixVhc4ER61KEkey5t5",
    "amount": 314000000,
    "amount (SMART)": "3.14",
    "exp": 3600,
    "id": "7c2666541a",
    "index_url": "https://electrum.org/r/index.html?id=7c2666541a",
    "memo": "this is a test",
```

(continues on next page)

(continued from previous page)

```
"request_url": "https://electrum.org/r/7c2666541a",
"status": "Pending",
"time": 1450175741
```

This command returns a json object with two URLs:

- request_url is the URL of the signed BIP70 request.
- *index_url* is the URL of a webpage displaying the request.

Note that request_url and index_url use the domain name we defined in *url_rewrite*. You can view the current list of requests using the *listrequests* command.

Open the payment request page in your browser

Let us open *index_url* in a web browser.



Fig. 65: Payment request page in a web browser

The page shows the payment request. You can open the smartcash: URI with a wallet, or scan the QR code. The bottom line displays the time remaining until the request expires.

This page can already be used to receive payments. However, it will not detect that a request has been paid; for that we need to configure websockets.

Add web sockets support

Get SimpleWebSocketServer from here:

git clone https://github.com/ecdsa/simple-websocket-server.git

Set websocket_server and websocket_port in your config:

```
electrum setconfig websocket_server <FQDN of your server>
electrum setconfig websocket_port 9999
```

<mark>⊗</mark> ● ■ <u>F</u> ile <u>W</u>	Elect allet	rum 2.5 <u>T</u> ools	i - defa <u>H</u> elp	ult_wa	llet							
History	Send	Recei	ve Ado	lresses	Contacts	Console						
Pay to	el	lectrum	.org									
Descript	ion th	is is a te	est									
Amount	31	140	mB1	C		729.56	USD					
Fee	0.	02986	mBT	C								
						S	end	Clear				
						S	end	Clear				
Invoices						S	end	Clear				
Invoices Expires		▲ [Request	Dr		S Descrip	end	Clear	Amount		Statu	s
Invoices Expires 2015	-09-19	 12:31 €	Request	or Im.org		Descrip this is a	end otion test	Clear	Amount 3 146),	Statu	s attente
Invoices Expires 2015	-09-19	F 12:31 €	Request	or Im.org		Descrip this is a	end otion test	Clear	Amount 3 140),	Statu	s attente
Invoices Expires 2015	-09-19	≜ F 12:31 €	Request	or Im.org		Descrip this is a	end otion a test	Clear	Amount 3 146),	Statu	s attente
Invoices Expires 2015	-09-19	≜ F 12:31 €	Requesta	or Im.org		Descrip this is a	end otion test	Clear	Amount 3 146),	Statu	s attente
Invoices Expires 2015	-09-19	▲ F 12:31 €	Request	or im.org		Descrip this is a	end	Clear	Amount 3 146),	Statu	s attente
Invoices Expires 2015	-09-19	← F 12:31 €	Request	or im.org		Descrip this is a	end btion a test	Clear	Amount 3 146),	Statu	s attente

Fig. 66: Wallet awaiting payment

And restart the daemon:

electrum daemon stop electrum daemon start

Now, the page is fully interactive: it will update itself when the payment is received. Please notice that higher ports might be blocked on some client's firewalls, so it is more safe for example to reverse proxy websockets transmission using standard 443 port on an additional subdomain.

JSONRPC interface

Commands to the SmartCash Electrum daemon can be sent using JSONRPC. This is useful if you want to use Smart-Cash Electrum in a PHP script.

Note that the daemon uses a random port number by default. In order to use a stable port number, you need to set the *rpcport* configuration variable (and to restart the daemon):

electrum setconfig rpcport 7777

With this setting, we can perform queries using curl or PHP. Example:

```
curl --data-binary '{"id":"curltext","method":"getbalance","params":[]}' http://127.0.
→0.1:7777
```

Query with named parameters:

Create a payment request:

```
curl --data-binary '{"id":"curltext","method":"addrequest","params":{"amount":"3.14",

→"memo":"test"}}' http://127.0.0.1:7777
```

🚪 History	🚿 Send	Receive	SmartNodes	🖧 SmartVote	SmartRewards	I Addresses	🔳 Coins	Contacts	1
Date	Description							Amount	Balan

Fig. 67: SmartCash Electrum Wallet

1.6.3 Android Wallet(deprecated)

SmartCash offers wallet for Android, this is the mobile versions of web wallet so have all features of web wallet.

https://play.google.com/store/apps/details?id=cc.smartcash.wallet

Android Wallet

1.6.4 iOS Wallet(deprecated)

SmartCash offers wallet for iOS, this is the mobile versions of web wallet so have all features of web wallet.

https://itunes.apple.com/us/app/smartcash-wallet/id1438215565?mt=8

iOS Wallet

1.6.5 Smartcard Wallet

Getting Started

The SmartCard is a way to hold SMART on a physical card. You don't need a phone to hold your SMART. If you have ever had a problem with using a phone inside a building you will get why this is ideal. Only the merchant needs the internet connection. As simple to use as a credit or debit card but not controlled by any 3rd party processor. And not a preloaded card using a visa fiat system... this is real crypto to crypto.

The easiest way to install the SmartCard for Android is from the Google Play Store.

GET IT ON
Google Play

Installation

Creat the SmartCard https://card.smartcash.cc/ All in One - SmartCard Guide

Deposit

Scan QR code to show SmartCash address and send smartcash to it.

Sending

Show your smartcard at merchant. Merchant will use smartpay app to scan smartcard. You will enter the PIN to confirm transactions.



Viewing transactions and balances

Using explorer to check your transactions details with your smartcash address explorer

Security FAQ

Breaking QR if stolen: We use BIP38 encryption that is executed in each own device (browser and app in the future). Encryption generates for each card: random bytes and unique decryptionKey (GUID) hashed with Keccak-256.

Private key then is encrypted with AES256 with decryptionKey hash and random bytes. So even if one QR Key is "cracked", all other Encrypted Keys are safe because random bytes and decryption key (GUID) are unique and almost impossible to duplicate. Each QR decryption takes about 1 second in modern CPU, David calculated how many years would take if we could do 1000 hashes per second. $((32^{16}) / 100\ 000\ 000) *$ seconds = 383 093 658 years to crack each QR

SmartPay Tempering: first you should do business with whom you trust and second tempering with the SmartPay or POS in the future, it takes at least the same tempering effort than any other payment terminal out there. Communication between SmartPay and Card API is encrypted and with a secret authorization which only the official application build has. That's why Card Name verification allows to guarantee Card API communication is authentic. If people have an hidden camera recording the device, they could record the same way any bank card front/back.

Payment Protection: on top of this, with SmartCard you can activate and reset your PIN on the fly, which you can't do with most bank cards. So you could transact and change PIN right after or just activate/deactivate card at payment time.

Security in SmartPay

Protection of Card Data: Captured card data (QR/NFC) is not stored in the device and SmartPay has protections against common attacks.

PIN Security: We have built a secure customised PIN pad that reinforces keylogger protection and keys are randomised with each use.

SmartPay Authenticity: While paying card holders should check in the PIN screen that "Verify Card Name" matches the card name.

Mandatory Updates: SmartPay requires to be using it's latest released version, ensuring SmartPay is always updated.



Smartcard Wallet

FUTURE POTENTIAL OF SMARTCARD...

Debit and credit card payments have arguably become the dominant way for many to carry out most, if not all, of their daily transactions. Adoption of card payments became increasingly simple for smaller businesses, and now quick and affordable smartphone-powered solutions are already a reality. SmartCard emerged offering to solve many of the problems facing traditional systems. SmartCard is blockchain-based solution that aims to improve on the digital payment card model. Our main goals are to streamline crypto transactions and make them practical for use in real-life payment scenarios. At the same time, we aim to drastically reduce fees and confirmation time frames over those found in current solutions.

SOLVING MAJOR CHALLENGES WITH A SIMPLE SOLUTION

As they currently stand, both traditional cards and crypto payment solutions leave a lot to be desired. Transaction fees of over 3% for some cards can add up to substantial losses over time for businesses of all sizes. For even small businesses, it's normal for more than \$50,000 a year to be lost in processing fees.

In addition to this, both buyers and sellers must have access to major banks and their services in order to facilitate these types of transactions; a major problem in many parts of the world. Also, traditional services inherently have long confirmation times of up to several days, which can themselves lead to other issues. Cryptocurrencies have indeed proposed solutions to many of these challenges, as it is easy to find many projects today that offer instant transactions and virtually no fees. But, this doesn't change the fact that using this new technology in its current form is like trying to send an email in the 80s – too time-consuming to setup and operate to be practical for most everyday applications.

Using their new feature called the SmartCard, the developers of SmartCash are looking to improve upon and/or eliminate entirely the present limitations of payment cards. SmartCard is the crypto alternative to the debit card which uses the SmartCash blockchain to easily initiate and verify transactions of any size and frequency. As with many other cryptocurrencies, the fees are right around 0%; it costs just fractions of a cent to process a transaction, regardless of the amount. The payments will be instantly confirmed at the time of purchase, meaning there is no delay between authorization of the payment and the transfer of the actual funds. In a way, the purchases made with SmartCard combine the best of cash, card, and crypto payments. The funds are moved between parties instantly, like cash, but the transaction is authorized by the buyer with a simple code that can be either digitally stored on a smartphone or printed and carried separately as a physical QR code. This makes the payment process just as convenient as regular card payments, and far easier than having to send cryptocurrency manually from a mobile app.

HOW IT ALL WORKS

There are just two main applications: the card app for the consumer and the Point of Sale (PoS) app for the merchant.

The card app allows anyone to create a SmartCard with just a few taps. Each card has an associated public address which can be used to load more funds onto the card from any wallet or exchange that offers SmartCash. Also included into each one is a QR code which can be scanned by the PoS application from either a phone screen or from a simple piece of paper. Once loaded through this address, the card works very similar to a pre-paid debit card.

The PoS app is equally responsible for this convenience. The merchant only has to specify the transaction amount in the local currency, and the app automatically initiates a transaction in the equivalent amount of SmartCash. The merchant uses the app to scan the code and requests the buyer to enter a confirmation pin. After entering it on the merchant's smartphone, the app connects to the blockchain to submit the transaction. If the code and pin match, and there are enough funds in the card address, the transaction is instantly verified and recorded on the blockchain. The process of scanning a card and entering a pin to confirm is already well familiar to current debit card users, and so it would be easy for everyone to get used to.

A REVOLUTIONARY PAYMENT SYSTEM EVERYONE CAN LOVE

This ease of use and familiarity present the key advantage of this system over other crypto payments, which require both parties to be online for a transaction. Once the card is created and loaded, only the merchant is responsible for having a live connection to the internet and the consumer can take the funds anywhere without worrying about a dead battery or lack of signal in remote areas. No need for fumbling with mobile wallet apps, nor any memorization beyond the simple numerical pin just like the ones used for debit purchases. In addition to this, it also offers the advantage of being usable in places where traditional banking services are only partially available, or even absent entirely. Whereas other mobile payment solutions require bank accounts for both parties to send/receive the funds, this one uses the SmartCash blockchain instead and is thus accessible to anyone who wishes to use it.

Naturally, these benefits would also be quite attractive to buyers and sellers in more developed nations as well. Other crypto solutions fall somewhat short of their promises. Crypto debit cards suffer from the same fees and transaction delays as any other current plastic card, and sending money through current mobile wallets requires a lot of time and effort to make the payments. Compared to these options, SmartCard payments are in a league of their own in convenience and speed. Add to this the fact that the SmartCash community is actively seeking, recruiting, and sponsoring business owners and entrepreneurs globally, and it is easy to imagine how the SmartCard could increasingly gain global adoption across a wide range of markets and economies.

1.6.6 Paper Wallet

The SmartCash Paper Wallet generator allows you to generate, encrypt and secure the keys to a single SmartCash address on a clean computer without ever connecting to the internet. Perfect for long term secure storage.

Introduction

A paper wallet is a method of storing a private key to access funds stored on a single address. It can be generated on a computer that has never been connected to the internet, and printed out for air-gapped offline storage. As such, they are suitable for storing large amounts of SmartCash, but care must be taken not to lose the private key, since there is no way of recovering funds if it is ever lost. To use the key, it must be imported or "swept" into an online wallet and should not be used again. Paper wallets are extremely secure but somewhat inconvenient for everyday use compared to hardware wallets, which also offer a high degree of security.

Paper wallets use random user and machine input to create a set of keys/addresses which you then print. You can never regenerate a paper wallet once you turn off the machine. What you print is all you get. For this reason, paper wallets are somewhat vulnerable and require special care because they can get damaged, lost, destroyed or stolen. Even if you encrypt them with BIP38 (which you should), a sufficiently motivated adversary (e.g. robbery/home invasion) could bypass this encryption using the proverbial "\$5 wrench attack".



Fig. 68: The \$5 wrench attack. Credit: xkcd.com

Nevertheless, together with appropriate planning, paper wallets are a highly convenient and user-friendly way to store SmartCash long term.

Security

While you can create a paper wallet using a machine that is connected to the internet, wallets that will be used to store significant funds should be generated using an offline computer running a single-use operating system to ensure that all generated data will be permanently wiped from memory once the process is complete.

A simple method of doing this is to burn a live Linux CD. Ubuntu Desktop is recommended because it will have the most drivers and is simple to use, while Tails and Kali Linux are popular choices for extremely strong security. Booting from an actual CD is most secure since it is mounted read-only, but a USB stick is generally fine as well. Both laptops and desktops can be used if you can ensure that all networking hardware is disabled when you get to the stage of actually generating your keys.

Boot from the CD and download/install your tools (or download them ahead of time to a USB drive). Disconnect from the internet, generate your keys/addresses/printouts, and power off the machine. You are now the only person with access to these addresses.

Death plan

Whichever type of cold storage you choose, make a plan to pass on the necessary data to regenerate the keys to your loved ones in the event of an accident - it will happen to us all eventually. Write down your paper wallet BIP38 decryption password or brain wallet passphrase. Then write down instructions on how to use it, and keep them separate with a clear procedure on how they can be accessed when necessary.

Tools

A SmartCash paper wallet can be generated in several ways.

- Using the generator at https://paperwallet.smartcash.cc/
- Offline using the SmartCash Paper Wallet source code from GitHub at https://github.com/SmartCash/ PaperWalletGenerator
- Offline using the same code which powers both sites, by viewing the GitHub project or downloading directly

Since the source code for all three options is largely similar, this guide will use https://paperwallet.smartcash.cc/ as an example. The websites listed here run entirely in your web browser without sending any of the data generated to an external server, but the most secure option is to download the wallet generator and run it on a computer with a freshly installed operating system that is not connected to the internet.

This guide is based on the guide available from https://walletgenerator.net. Please donate if you find this project useful!

Address generation

Go to https://paperwallet.smartcash.cc/ in your web browser (or open index.html if you downloaded the wallet generator). Select your language and choose SmartCash as the currency if necessary. The following screen will appear:

Open Source JavaScript Client-Side SmartCash Wallet Generator					
One Address Paper Wallet	Multi Address Paper Wallet				
Generating SmartCash Address MOVE your mouse around to add some extra random OR type some random characters into this textbo	ness X				

Fig. 69: The SmartCash Paper Wallet Generator at paperwallet.smartcash.cc

Some random data is required to ensure the generated address and key are unique. Move our mouse around and/or type random characters into the text box until the process reaches 100% and the following screen appears:

Once your public address and private key (shown in Wallet Import Format or WIF) are visible on the **Single Wallet** tab, you should immediately click **Print** to print the data and store it securely. If you leave the page without somehow recording the smartcash address and private key, all data will be irretrievably lost, together with any funds you have sent to the address.

Sending to a paper wallet and viewing the balance

You can send SmartCash to a paper wallet address in the same way as to any other SmartCash address. See the documentation for your wallet if you do not know how to do this. For this example, 0.05 SmartCash (minus transaction



Open Source JavaScript Client-Side SmartGash Wallet Generator

28%	28%
Generating SmartCash Address MOVE your mouse around to add some extra randomn OR type some random characters into this textbox eca66c7f53a11e9107a328bf5d57b4e8431ff2e7bec7043 1120e1d51ef4cb17aeb49786ef3a81580cb4eb5c37c5f4c6 79d11a0c849fa7025fc4770e1a94607120955083c9a0ca6 a927e1f780a7bdd13de5bf08231085ff53b3dda9e5b020f8 900b354b57448efae31dc0a5dadcd4ba87be0550046cb781 6f13718e9610cd10a5067dbebf135a82968175cbd78a80c0 9d835b1a53a5fa4b2f655ae2f8cb7445c3fa78e30627d88 ce2712b18330069	ess 28% a01255ed68177dd43b1241c affebcd4dedbe6599a63455 bfbcff8c75eecdce75250b3 7d8484bb84d63f7664be001 e1ab6375911008cbd54ce05 b5d2f98071a814b3aa71980 9326e0dddf9de2823b826df
. · · · ·	

Fig. 70: Generating randomness for the SmartCash Paper Wallet Generator



Fig. 71: A SmartCash address and private key generated using SmartCash Paper Wallet Generator

fee) has been sent to the paper wallet address. Anyone with knowledge of the public address is able to see the balance of the wallet using a block explorer, but only someone with knowledge of the private key can access the funds. You can make as many deposits and send as many coins to the same address as you'd like. Just make sure you test your wallet with small amounts first to learn how it works.

SmartCash Blocks Status Search for block	Status Search for block, transaction or address		·Height 847507 문문 Scan SMART •	
Block #847505				
BlockHash 41604fe16efb181edc05ca975501bef9032d7ea84d3418dd	óde86fc3ef3f2adbi	2		
Summary				
Number Of Transactions	2	Difficulty	349364849.04794055	
Height 847505 (Mainchain)		Bits	3its 1b00c4b2	
Block Reward	847 SMART	Size (bytes)	6406	
Timestamp Jan 16, 2	019 5:26:46 PM	Version	2	
Mined by		Nonce	2074565339	
Merkle Root 40282b04da3ecec483d080d0cbfa763c				
Previous Block	847504			
Transactions				
Ca589426386ff853ae2afd17d621214b9785f26a104646931a2edd7c9c7a408c			mined Jan 16, 2019 5:26:46 PM	
No Inputs (Newly Generated Coins)	>	SZYJS4H713hht3KoZaSVrexH8ufUjeJJcD	42.3561 SMART (U)	
		Unparsed address [0]	0 SMART (U)	
		10	ONFIRMATIONS 42.3561 SMART	

Fig. 72: Viewing the balance of the paper wallet using the SmartCash Block Explorer at explorer.smartcash.cc

Spending from a paper wallet

In order to access the funds stored on the paper wallet address, you will need the following:

- · The public address
- The private key in WIF

If you encrypted the wallet, you will additionally need the following to convert the BIP38 key into the WIF key:

- The encrypted private key in BIP38 format
- The passphrase you used to encrypt the key

Optional: Decrypt BIP38 key to WIF

If you encrypted your paper wallet, you will first need to decrypt the BIP38 key. You can skip this step if your private key was not encrypted.

Importing the private key to your live wallet

When you are ready to spend the balance on the paper wallet, you will need to import the private key used to control the address printed on the wallet into another SmartCash wallet that is connected to the internet. We will use the SmartCash Core Wallet in this example, although SmartCash Electrum and mobile wallets are also supported. Open SmartCash Core Wallet, click **Settings** and **Unlock Wallet**. Enter your wallet passphrase. Then click **Tools** and select **Debug Console**. The console appears. Enter the following command:
importprivkey <your in="" key="" private="" wif=""></your>	
importprivkey VGCRAm15XoqtjkLheyq5JJxgbcfdu63mTPdBfVEq7AVFtb6yyCoC myNewAddress	1

Fig. 73: SmartCash Core Wallet importing a private key

This process requires rescanning the entire downloaded blockchain for transactions involving this address, and will therefore take some time. Be patient. Once the process is complete, any transactions involving the imported address will appear in your list of transactions. If you use Coin Control, you can also enable or disable the address for spending.

Since the paper wallet public address still holds the funds, it can also be imported again into a second wallet if it is not destroyed. It is recommended to transfer the balance from the paper wallet to an internal wallet address or another address where you have exclusive control over the private key. This will prevent a third party from obtaining unauthorised access to the same address from the paper wallet before you do. You can then spend your balance as usual.

Once the paper wallet is empty and you are sure it will not be receiving any further deposits, you can destroy the paper.



Fig. 74: SmartCash Paper Wallet

1.6.7 Hardware Wallets

A hardware wallet is a type of device which stores private keys for a blockchain in a secure hardware device, instead of in a database file such as wallet.dat used with common software wallets. This offers major security advantages over software wallets, as well as practical benefits over paper wallets. To date, there is no verifiable evidence of hardware wallets being compromised by viruses, and they are also immune to keylogger attacks that could be used to steal passwords to unlock the private keys used with software wallets.

Hardware wallets function by storing your private keys in a protected area of a microcontroller. It is impossible for the private key to leave the device in plain text - only the signed output of the cryptographic hash is ever transmitted to the device interacting with the blockchain, such as your computer or smartphone. Most hardware wallets feature a screen which allows you to independently confirm the address you are sending to is correct.

This section lists the most common commercial hardware wallets supporting SmartCash, although some other enthusiast projects may also be available.

Introduction

Hardware wallets offer you the security of storing your keys in secure device while still allowing you to make simple transactions through a web interface.

Trezor



Developed by Czech startup SatoshiLabs, the \$99 device is essentially a USB dongle designed to add an extra authentication layer to all outbound bitcoin transactions. Trezor has supported SmartCash since 2018.

By virtue of its design, Trezor can be used to sign transactions on 'unsafe' computers and is impervious to keyloggers and many other vectors of attack, so even if your host PC is compromised, the attacker has no way of getting your private key. That's also where the device gets its name, as 'trezor' translates into 'vault' in most Slavic languages, including Czech. A kind of 'vault' for your private bitcoin key, Trezor claims to use a number of clever tricks to maintain security even on compromised and unsafe machines.

- Site: https://trezor.io
- Shop: https://shop.trezor.io
- Wallet: https://wallet.trezor.io

It is also possible to operate a SmartCash Smartnode using your Trezor. See here for details.

Getting Started

Once you have bought your Trezor from https://shop.trezor.io or an authorized reseller, you will need a wallet to use it with. Trezor supports the following SmartCash wallets:

- Trezor Wallet (documentation)
- SmartCash Electrum Wallet (documentation)

This documentation describes how to get started using the official Trezor web wallet at https://wallet.trezor.io. Always confirm the URL is correct and SSL encryption is enabled when working with the Trezor Wallet. Follow these steps when setting up your Trezor for the first time:

- 1. Inspect the packaging for tampering. There should be two seals and the flaps should be glued shut. It should be impossible to remove the device without totally destroying the packaging.
- 2. Go to https://trezor.io/start/ and watch the video to introduce the concepts of a shifting PIN layout and recovery seed.
- 3. Go to https://wallet.trezor.io/ to begin the setup process.
- 4. If not already installed, install the Trezor Bridge application from https://wallet.trezor.io/#/bridge
- 5. Connect the Trezor to your computer when prompted.
- 6. If this is the first time you connect your Trezor, you will be prompted to install firmware. Click the **Install** button, wait for the download and confirm on the device.

- 7. When complete, the device will display a fingerprint. Verify that this matches the fingerprint shown on the screen. Note that this is hexadecimal and therefore not case-sensitive.
- 8. After verification is complete, disconnect and reconnect your device. Enter a device label on the screen that appears.
- 9. Enter and confirm a PIN by clicking on the squares according to the mapping shown on the device.
- 10. Your Trezor device will now display a sequence of 24 words on the screen. This is your recovery seed. Write the words down in the order they appear on the recovery card. Never store your recovery seed in any digital format, including photos or text.
- 11. Verify the seed against what you have written down and store it in a safe place.
- 12. You will be asked to enter your PIN again.
- 13. The Trezor Wallet will appear with a message that your device is ready for use. Your device name will appear on the device.
- 14. Switch to the SmartCash wallet using the menu at the top left. You can now use your Trezor to send and receive SmartCash.



Fig. 75: Trezor Web Wallet for SmartCash ready for first use

Ellipal



ELLIPAL Hardware Wallet is completely isolated from all physical and network connections. This means ELLIPAL cannot be connected to any devices or to any networks, completely protecting it from hackers aiming for your coins.

Without ever relying on USB or Bluetooth connection to work, ELLIPAL is the safest hardware wallet in the market.

ELLIPAL is an air-gapped device, it works flawlessly with ELLIPAL App via secure and verifiable QR codes.

- Site: https://www.ellipal.com
- Shop: https://www.ellipal.com/collections/all
- Wallet: https://www.ellipal.com/pages/hardwarewallet-new



Fig. 76: Ellipal Wallet support SmartCash

Soon @scashofficial will have support for hardware wallet @Trezor look at the video our first transaction #dev #blockchain \$SMART



Fig. 77: Trezor and Electrum SmartCash Wallet

1.6.8 Third Party Wallets

These wallets have been developed by external developers to support SmartCash. Many third party wallets support multiple different cryptocurrencies at the same time, or integrate instant cryptocurrency exchanges.

Introduction

The SmartCash protocol and many SmartCash products such as SmartCash NodeClient are entirely open source, which makes it easy for third parties to integrate SmartCash with their existing cryptocurrency wallet solutions. This section describes some of the third party wallets available and the functions they offer. Please note that SmartCash does not provide support for any of these wallets, and any listing here should not be considered an endorsement or recommendation. Contact the software vendor for support.

Atomic Wallet

https://atomicwallet.io



Atomic Wallet is a multi-asset custody-free wallet with atomic swap exchange and decentralized orderbook functionality. It provides a powerful, secure service that transparently and reliably allows users to reduce effort spent on managing and exchanging crypto assets.

Installation

All Atomic Wallet releases are available from https://atomicwallet.io - simply download and install the appropriate package for your system.

Documentation

Atomic Wallet offers detailed documentation of all functions at https://atomicwallet.freshdesk.com and a few quick links are also collected here:

- Getting started with Atomic Wallet
- How to install Atomic Wallet
- How to create a wallet
- Getting started with Atomic Swaps

Coinomi

https://coinomi.com



Fig. 78: Atomic Wallet Portfolio screen



Coinomi is an open-source multi-currency mobile wallet available for iOS and Android. Your private keys never leave your device, and strong wallet encryption guarantees that your funds are always under your control only. Instant exchange is available directly in the wallet through ShapeShift and Changelly integrations.

Download



Coinomi is available from the Google Play Store for Android and the Apple App Store for iOS.

Documentation

Coinomi offers detailed documentation of all functions at https://coinomi.freshdesk.com



Fig. 79: Coinomi wallet running on Android

Edge

https://edgesecure.co

Edge is a secure multi-currency wallet for iOS and Android. It offers a unique login system to store your encrypted HD seed on the cloud while still performing all sensitive operations requiring a private key on your device. Edge is fast and simple to use, allowing you to scan QR codes and sign transactions using your fingerprint ID or a simple PIN code. ShapeShift is also integrated to facilitate exchange between different cryptocurrencies.

Installation



Edge is available from the Google Play Store for Android and the Apple App Store for iOS.

Documentation

Edge offers detailed documentation of all functions at https://support.edgesecure.co and a few quick links are also collected here:



- Getting started
- How do I create a new wallet?
- How do I send money?
- How do I receive money into my account?
- What is ShapeShift and how does it work?



Fig. 80: Edge Welcome, Receive SmartCash and Balance screens

Ethos

https://www.ethos.io



The Ethos Universal Wallet allows you to store SmartCash and over 100 other cryptocurrencies. It features a single recovery phrase, known as the SmartKey, with which you can restore all balances on another device. Features include human-readable usernames, ShapeShift coin conversion and a portfolio tracker.

Installation





Ethos Universal Wallet is available from the Google Play Store for Android and the Apple App Store for iOS.

Documentation

Ethos offers detailed documentation of all functions at https://support.ethos.io and a few quick links are also collected here:

- How to Create and Ethos Account
- How to Create Your SmartKey and First SmartWallet
- How to Add Coins to a SmartWallet
- How to Receive Cryptocurrency to a SmartWallet
- · How to Send Cryptocurrency from the SmartWallet



Fig. 81: Ethos Universal Wallet on Android

Pungo

https://pungo.app/



Pungo Wallet is focused on ease of use, security, and especially on maintaining the end user's financial independence as it's a noncustodial wallet. Our company never has access to the user's funds. Pungo Wallet offers a multicrypto wallet which supports a growing amount of coins. We provide a built-in fiat gateway to be able to buy and sell from the wallet through convenient methods. We also provide a debit card which can be ordered and loaded from the wallet interface. Our goal is to become a gateway for blockchain based services and a reference wallet for those users searching for independence from mainstream financial solutions by providing a full set of convenient services seamlessly integrated within the wallet.

Installation



Pungo Wallet is available from the Google Play Store for Android

Documentation

Pungo Wallet offers detailed documentation of all functions at <https://pungo.app>_

Fig. 82: Pungo Wallet on Android



Fig. 83: Edge Wallet

1.6.9 Web Wallets(deprecated)

https://wallet.smartcash.cc/, which provides a secure web interface to the SmartCash blockchain while leaving you with full control of your private keys.

Web Wallet (deprecated)



WebWallet is a web interface to the SmartCash blockchain. It is explicitly not an online wallet, meaning you maintain control over your private keys at all times. Unlike many other light wallets, Web Wallet also supports advanced SmartCash features such as instantpay, voting, smartvault, smartcard, buy or convert.

Web Wallet offers complete and detailed documentation for all functions.

- All about web wallet
- WebWallet New Account Registration
- How to Send Smart from your WebWallet
- · How to import privatekey from webwallet to desktop wallet and viceversa

(E) SmartCash	Language 6	EN Ψ Σ 0.5	996 \$ 0.0129 Tor	1 :Login: Jan 15, 2019, 9:47:47 AM
MAIN	Here are your 5 smarte	cash addresses. You c	an use them to receive and send smartcasl	h
💎 Overview		🧭 Buy Sr	martCash	
► Send				
1 Receive				
→ ⁺ Transactions	ο Σ 0.978		Σ 0.018	_
E Cards	SPENDING		SMART REWARDS	
🌢 SmartVault	Address	-	Address	
🖉 Buy or Convert	ScWnALHW9Mhf5R3SUuWWføCubø9GGTtTD	' D'	Sb21D44JDk8DUgbMR5dPq1KkR38vKSEpym	'

Fig. 84: My SmartCash Wallet

1.6.10 Wallet Guides

Documentation in this section describes common tasks and questions relating to all wallets.

Wallet Recovery

Long-time users of cryptocurrency sometimes find old wallet files on USB drives or cloud storage that they have forgotten about. Others may have a backup, but can't remember the software they used to create it, or have forgotten the password. Other users may have an old version of SmartCash NodeClient that no longer works because the network has upgraded. This documentation is intended to help these users restore access to their funds.

Determining the backup format

The first step is to determine the format of your backup. In most cases, this will either be a file, probably named *wallet.dat*, or a phrase of words. In some cases, you may have stored the private key for a SmartCash address directly. The following list shows the possibilities and methods to restore your wallet in order of probability.

- · Backup is stored in an older version of SmartCash NodeClient that no longer works
 - Follow instructions for restoring wallet files using SmartCash NodeClient
- Backup is a file
 - If file name is similar to wallet.dat, try to restore using SmartCash NodeClient

- Backup is a phrase of words
 - If 12 words long, try to restore using SmartCash Electrum wallet
 - If 13 words long, try to restore using SmartCash Electrum wallet
 - If 12, 18 or 24 or 25 words long, try to restore with the *hardware wallet* you used to create the recovery phrase
- Backup is a long string of random characters or a QR code
 - If 34 characters long and starting with S, this is a public address and cannot be used to restore access to lost funds. You need the private key.
- If 51 characters long and starting with V, this is a private key in WIF, import using SmartCash NodeClient

Once you have determined your backup format, follow the links to view the restore guide for that format.

File Backups

SmartCash NodeClient

One of the most common wallet backup formats is a *wallet.dat* file from SmartCash NodeClient wallet. Before you begin, make absolutely sure that you have a copy of this file stored somewhere safe in case the restore process accidentally corrupts your wallet file! In most cases, *wallet.dat* backups will also be protected by a password, which you will need to know to regain access to your SmartCash funds. If you already have SmartCash NodeClient installed, first ensure it has been updated to the latest version by clicking **Help > About SmartCash**. Compare this with the latest available version of SmartCash NodeClient on the website as follows:



Update SmartCash NodeClient to the latest version according to the *installation instructions*. If you have only a wallet file and no existing installation of SmartCash NodeClient, simply install SmartCash NodeClient according to the *installation instructions* and start it once to create the SmartCash folder. Then close SmartCash NodeClient and copy the *wallet.dat* file you want to restore to the SmartCash folder in the location specified below, replacing or renaming the existing file.

Plat-	Path to data folder	How to navigate
form		
Linux	~/	Go to your home folder and press Ctrl+H to show hidden files, then open
		.smartcash
ma-	~/Library/Application	<pre>Press Shift + Control + G, type ~/Library/Application Support,</pre>
cOS	Support/	then open SmartCash
Win-	%APPDATA%	Press Windows Key + R and type %APPDATA%, then open SmartCash
dows		

) C	https://smartcash.co	/wallets/#nodeclient			
				N	DDE CLIENT
	This Node Client requires the blockchain download which can take hours on slow machines. It is also only inte assist in speeding up the syncing of your Node Client read more <u>here</u>				
	Disclaimer: Do not menu or by closing external drive. D Al	move your wallet.d. the wallet and waitir o not shutdown you ways exit the wallet	at with you ng about a Ir compute from the r	ur wallet ru minute to er with wall menu or clo	nning. Backup your wallet by using the backup wall close and copy the wallet.dat file. Always keep at le et running or kill the wallet process. These can cau se x and wait for it to shutdown before turning off
	🐮 Windows (64bit)	🗱 Windows (32bit)	é OSX	∆ Linux	
	smartcash <mark>-1.2.6-</mark> win64-setup.exe				
		64bit			
					14.5MB
		39fa	4e383a52c	Bii 8852a61c2b	naries – SHA256 Checksum 77a63de753cb7b5b1cc1172b8b61e56f72c4b95464

Fig. 85: Comparing the installed version of SmartCash NodeClient with the latest version available on the website

To repair a broken installation, navigate to the SmartCash folder and delete all *.log* and *.dat* files except *wallet.dat*. The following files can be safely deleted:

- banlist.dat
- db.log
- debug.log
- fee_estimates.dat
- version.dat
- snpayments.dat
- netfulfilled.dat
- peers.dat

Leave .conf files and the folders (such as backups, blocks, chainstate, etc.) intact, since they will help you get started faster by providing a copy of the blockchain and your settings.

Now open SmartCash NodeClient and wait for blockchain synchronization to complete. Your wallet will be restored/upgraded and all balances should be displayed. You should ensure you have the correct password by trying to unlock your wallet from **Settings > Unlock Wallet** to make sure you can actually create transactions using your balances. If you have any problems with your balance not appearing, try to force a rescan of the blockchain by going to **Help > Debug Windows >Wallet Repair** and selecting **Rescan blockchain files**. **Rebuild index** may also help. SmartCash NodeClient will restart and perform a full scan of the blockchain.

At this stage, recovery is complete and you should make another backup using **File > Backup Wallet** or following the instructions *here*. If you have any further problems, try asking on the discord, Reddit or the

Recovery Phrases

12/13-word phrase on SmartCash Electrum

Ensure you are using the latest version of SmartCash Electrum according to the installation instructions *here*. Smart-Cash Electrum supports multiple simultaneous wallets, so you can safely restore to a new wallet file without losing your old wallet. Click **File > New/Restore** and enter a file name to store your new wallet. Then select **I already have**

a seed and enter your 12/13-word recovery phrase. Enter a new password for your wallet and click **Next** to recover your addresses from the recovery phrase.

Hardware wallet recovery phrases

If your 12, 18 or 24-word recovery phrase was generated by a hardware wallet, follow these instructions:

- Ledger Nano S
- Trezor

Private Keys

Most wallets offer a function to import an address from a private key, see the documentation for your wallet for specific instructions. While private keys can be stored in many ways, in this example we will work through the process of restoring a private key from a paper wallet using SmartCash NodeClient. If you only have a QR code and not the key, use a barcode scanning app (Android or iOS) to read the code first.

First, start SmartCash NodeClient and unlock your wallet by selecting **Settings > Unlock Wallet**. Enter your password, then open the debug console by selecting **Help > Debug Windows**. In the console, type the following, replacing the example private key with your key:

```
importprivkey VrPQWnMrh3oWLtZrzt1zLRSCVyuBbwnt7fRBXPp2EwcPhtzXSzp
Importing a private key using the debug console in SmartCash NodeClient wallet
```

SmartCash NodeClient will rescan the blockchain for transactions involving the public address of this key and enter the transactions and balance in your wallet.

Forgotten Passwords

In most cases, if you selected a strong password and have forgotten or lost it, there is practically no hope of recovery. The encryption used by the SmartCash wallets is extremely strong by design, and a well-chosen password should defeat most brute force cracking attempts. If you can recall some details of the password, particularly its length or sequences of characters that may be included, then brute force password cracking techniques may be worth attempting. Several services exist to do this, or you can attempt it yourself. Because SmartCash NodeClient is based on Bitcoin Core, most approaches to apply brute force to crack a Bitcoin wallet will also work for SmartCash wallets.

- Wallet Recovery Services
- BTCRecover

Signing and Verifying Messages

SmartCash wallets function by securely storing the private keys used to create transactions using publicly visible addresses on a distributed ledger called a blockchain. In some cases, it may be necessary to verify an address to demonstrate control of the funds that it holds, that you can receive using the address, or even to verify your identity to an exchange that has previously seen a transaction from that address. To avoid phishing attacks, take great care when sending signed messages to parties you do not trust, since ambiguously worded messages could be used to impersonate your identity.

Signing messages requires a private key. Verifying messages requires a public address. In this way, you can write an arbitrary message and generate a signature that could only have been created by a user with control of the private key for this address. Any user with the message, the public address and the signature can easily verify that only you could

have created the signature. Messages are not stored on the blockchain. Instead, you must send these three text snippets (message, address and signature) to your intended recipient using another communication channel, such as email. As an example, you can test the following message and signature and verify that it was signed by the given address:

Message: Documentation test signature Address: SP3WsLqWkmeyWikLfYjzBo7p8o4wWFZr1A Signature: IKaEt7MTb4Y57WIde5TLUkm1vbVsjk/vGgTsqcVOTmXXQ6QA0/WQtxqPURH2qZG27YR2VE0=

Signing a Message

The following screenshots show how to sign a message using the SmartCash Core wallet (**File > Sign message**), Trezor web wallet (Sign & Verify > Sign message) and DMT (**Tools > Sign message with HW for current masternode's address**):

Verifying a Message

The following screenshots show how to verify a message using the SmartCash Core wallet (File > Sign message) and Trezor web wallet (Sign & Verify > Sign message):

1.7 Earning, Saving and Spending

SmartCash is designed from the ground up to function as digital cash. This documentation discusses how and where SmartCash users can manage all of their personal finances using SmartCash.

1.7.1 Earning

Receive payment from your employer in SmartCash directly. You can then withdraw your wage to any SmartCash wallet for spending or saving.

You can earn SmartCard by Proposal or get Tips from friend Tips bot

You can also earn SmartCash via Mining and SmartNode

1.7.2 Saving

You can save SmartCash and get reward via SmartReward

1.7.3 Spending

Merchant Directory

SmartCash can be spent in hundreds of stores and services both online and in physical locations.



Find SmartCash lists businesses around the world accepting SmartCash, sorted by category. It's easy to add your business to the list, and also features a short introduction for new SmartCash users.

SmartCard



https://card.smartcash.cc/

The SmartCard is a way to hold SMART on a physical card. You don't need a phone to hold your SMART. If you have ever had a problem with using a phone inside a building...you will get why this is ideal. Only the merchant needs the internet connection. As simple to use as a credit or debit card but not controlled by any 3rd party processor. And not a preloaded card using a visa fiat system...this is real crypto to crypto.

SmartPay

https://business.smartcash.cc/

Accept SmartCash as a payment option in your business with zero fees using a simple SmartCash app. The SmartPay app is available for use anywhere in the world, all it requires is access to the internet.

This app will allow you to:

- Accept payments instantly from a customer's SmartCard's using SmartCash's Instantpay technology.
- Receive payments from the customer to your business with no 3rd party.
- Manage different store locations with individual SmartPay staff logins.
- Allow optional tipping for staff on all transactions.

Have peace of mind with SmartPay's security features:

- · SmartPay processes transactions without access to business funds
- Private keys are created by you or generated in the browser
- · Divide payments into different wallets with HD Wallet or Multi Address
- Decentralized payment system gives you full control of your funds.

Start accepting cryptocurrency payments in your business with 0% fees, and download the SmartPay app today! Visit https://business.smartcash.cc to learn more.

Debit Cards

Debit cards work by prepaying in SmartCash to load the account, then withdrawing cash from an ATM or spending online or anywhere debit/credit cards are supported. The SmartCash is either exchanged at the time of purchase or in advance. The rapidly evolving approach to regulation of cryptocurrencies such as SmartCash and instant exchange solutions such as ShapeShift means that availability of debit cards cannot be guaranteed in any or all jurisdictions. Check with the following providers for updates on the availability of SmartCash debit cards.

Disclaimer: This list is provided for informational purposes only. SmartCash Team is not liable for any funds transmitted in error to these providers, or for the accuracy of information on this page.

Pungo.krd



https://pungo.krd/home/

A prepaid debit card that can be acquired in many places without an ID and with no bank account linked. It can be used by anyone residing legally in the EU once the owner gets verified.

1.7.4 Tax

Taxation law is different depending on where you qualify as a resident for tax purposes. The following services are available to help you calculate your tax obligations.

- https://cointracking.info
- https://bitcoin.tax

1.8 Getting Started

SmartCash welcomes new merchants and supports integration through a standardised onboarding process. It's easy to begin accepting payments in SmartCash and enjoy the following benefits:

- · Settlement within seconds and clearance within minutes
- · Ability to accept payments from any market around the world so help you grow your customer base
- · Irreversible transactions to prevent fraud
- · Advanced privacy for both customers and merchants
- · Lowest fees in the industry
- Pos Solutions and Ecommerce Plugins

How to integration SmartCash is available in English on SmartCash for Merchants.

Many major merchants accept SmartCash - check out Taco or Boralatour for examples of what merchant integration can look like. Once you are up and running accepting SmartCash, consider adding your business to the directory maintained at SmartCash Merchant Directory for increased visibility.

1.8.1 Payment Processors

This section lists known payment processors supporting SmartCash and the business platforms they support. Please conduct thorough research before choosing a payment provider to ensure your needs will be met.

For more advanced payment processing needs, such as for high risk merchant accounts in industries challenged with high levels of chargebacks, it is recommended to contact SmartCash partner Mycryptocheckout. or Coinpayments. or Anypay.

Point of Sale

A range of Point of Sale systems are available. Many function as an app or simple website serving a checkout interface and QR code generator, while others support custom features such as NFC or a rewards scheme. SmartPay and SmartCard (*Getting Started*) are particularly widespread.



SmartCard

https://card.smartcash.cc/

The SmartCard is a way to hold SMART on a physical card. You don't need a phone to hold your SMART. If you have ever had a problem with using a phone inside a building...you will get why this is ideal. Only the merchant needs the internet connection. As simple to use as a credit or debit card but not controlled by any 3rd party processor. And not a preloaded card using a visa fiat system...this is real crypto to crypto.

SmartPay

https://business.smartcash.cc/

Accept SmartCash as a payment option in your business with zero fees using a simple SmartCash app. The SmartPay app is available for use anywhere in the world, all it requires is access to the internet.

This app will allow you to:

- Accept payments instantly from a customer's SmartCard's using SmartCash's Instantpay technology.
- Receive payments from the customer to your business with no 3rd party.
- Manage different store locations with individual SmartPay staff logins.
- Allow optional tipping for staff on all transactions.

Have peace of mind with SmartPay's security features:

- · SmartPay processes transactions without access to business funds
- Private keys are created by you or generated in the browser
- Divide payments into different wallets with HD Wallet or Multi Address
- Decentralized payment system gives you full control of your funds.

Start accepting cryptocurrency payments in your business with 0% fees, and download the SmartPay app today! Visit https://business.smartcash.cc to learn more.

All in One - SmartPay Guide



Anypay

https://anypay.global/

Anypay is Fast, Private, No-Fees, Multi-coin Point of Sale.

Online Stores

Due to the wide range of platforms for online stores, the following table is intended to help you select an appropriate payment processor for your existing store.

CoinPayments Mycryptocheckout

SmartCash Ecommerce Plugins

Make your website run Wordpress become Ecommerce store with WooCommerce Plugin

SmartCash WooCommerce Plugins

WooCommerce and CoinPayments.net and Mycryptocheckout

If your online store is built on WooCommerce, you can simply install CoinPayments as an additional payment gateway and immediately begin accepting SmartCash. This guide assumes you have already installed Wordpress, installed WooCommerce and created at least one product in your store. A video of the process to install the CoinPayments payment processor is also available.

In your WordPress administration backend, select **Plugins -> Add New** and type "coinpayments.net" into the search box. A plugin named **CoinPayments.net Payment Gateway for WooCommerce** should appear. Click **Install Now**

to install the plugin. Alternatively, you can download the plugin from the WordPress website as a zip file and upload it using the **Upload Plugin** button. Once the plugin is installed, click **Activate** to begin configuration.

🛞 👚 Dash Documentation Shop 📮 0 🕂 New Howdy, straphy 🚡				
🖚 Dashboard	R8 Dashboard Add Plugins Upload Plugin			
✤ Posts ✿ Media	Search Results Featured Popular Recommended Favorites	Keyword • coinpayments.net		
Comments	CoinPayments.net Payment	Stats Dashboard for Coinpayments		
Products	More Details This plugin implements a payment gateway for WooCommerce to let buyers pay with Bitcoin, Litecoin, Ripple, and	A plugin to display stats from coinpayment.net		
Plugins Installed Plugins	other cryptocurrencies via CoinPayments By CoinPayments.net	Catal Updated: 1 year ago 20+ Active Installations Untested with your version of WordPress		
Add New Editor	まます または (4) Last Updated: 3 months ago 2,000+ Active Installations ✓ Compatible with your version of WordPress			
Lusers		2 items		
 Conapse menu 	Thank you for creating with <u>WordPress</u>	Version 4.9.5		

Fig. 86: Installing the CoinPayments.net WooCommerce plugin

Next, go to CoinPayments.net and sign up to create an account. Once you are logged in, go to Account -> Coin Acceptance Settings and enable SmartCash, as well as optionally entering a withdrawal address. Next, go to Account -> Account Settings and copy Your Merchant ID from the Basic Settings area into a text file. Then navigate to the Merchant Settings section and enter a long, random series of characters for the IPN Secret. Copy this code to your temporary file as well.

Back in the WordPress plugins section, click the **Settings** button for the WooCommerce plugin and navigate to **Check-out -> CoinPayments.net** section. Ensure the CoinPayments plugin is enabled here, then enter the **Merchant ID**, **IPN Secret** and **Description** in the appropriate fields as shown below. Click **Save Changes** when you are ready.

🚊 Coinhgreants 🚍	🔞 😤 Dash Document	ntation Shop 📮 0 🕂 New Howdy, strophy 😭 🏠
Account Settings	🍘 Dashboard	Help ¥ General Products Shipping Checkout Accounts Emails API
Construction of the second secon	 ≁ Posts ♀ Media ● Pages ♥ Comments 	Checkout options [BACS Check payments Cadh on delivery PayPal CoinPayments.net CoinPayments.net Completes checkout via CoinPayments.net
en para para de la calquera de la ca	Wee WooCommerce Orders Coupens	Enable/Disable Imable CoinPayments.net Title CoinPayments.net
Colorbymonis Account Settings	Reports Settings Status	Description Pay with Dash via CoinPayments.net
C O E Paracterization O E Paracterization O O O O O O O O O O O O O O O O O O O	Products Appearance	This controls the description which the user sees during checkout. Merchant ID
This taxatir work Max. If it is let us, our open contracting the maximum of the maximum of the maximum of the protocol of the maximum of themaximum of themaximum of the maximum of themax	 Fragms Users Tools Settings 	IPN Secret 지금 이상 가장 가장 수상 가장 수상 이상 수상 가장 수상 수상 가장 수상

Fig. 87: Configuring the CoinPayments.net WooCommerce plugin

Your customers will now see an option to pay with SmartCash when completing the checkout process for an order. The payment will be processed by CoinPayments.net, and you will receive emails detailing each purchase procedure. You can choose how often you want to withdraw your payments, to which SmartCash address and various other options in the CoinPayments administration section. See the CoinPayments Documentation or Merchant Tools for more information.



1.8.2 Installation Examples

This section contains worked examples of how to install, configure and process your first payment using the payment processors listed in this documentation.

https://find.smartcash.cc/a-guide-to-accepting-smartcash-payments-for-your-online-store/



Fig. 88: Completing payment through the CoinPayments.net payment processor

https://find.smartcash.cc/for-merchants/brick-and-mortar/

1.9 Administrative Processes

It's easy to get started integrating SmartCash, but you will need to make some decisions about whether you plan to convert your income earned in SmartCash into your local fiat currency, or if you prefer to hold some or all of it in SmartCash. Most payment processors offer a range of fiat conversion options, although various fees and limits may be applicable.

1.9.1 Onboarding Process

New merchants typically go through the following steps when joining the SmartCash ecosystem:

- 1. Set up a SmartCash wallet
- 2. Identify an appropriate payment processor
- 3. Decide on how and when to convert funds
- 4. Implementation and testing
- 5. Release and marketing
- 6. Integration on Find SmartCash

1.9.2 Promoting SmartCash

A wide range of ready-to-go visual products are available to help you promote SmartCash as a payment method to your customers. This includes promotional graphics and stickers, fonts for consistent visual design and guidelines on how to use the SmartCash visual identity. See the *Marketing* section for more information.

The reduced fees may also offer an additional incentive for your customers to pay with SmartCash, particularly in businesses with high cash handling fees or where it is necessary to add a fee to process credit card transactions.

1.9.3 Currency Conversion

Cryptocurrency is a relatively recent development, and rapid development in the ecosystem coupled with various barriers to access and heavy trading mean that fiat-denominated value is subject to considerable fluctuation. As a merchant, you will need to make decisions about how much of your income taken in cryptocurrency should actually be held in cryptocurrency, and how much should be converted back to a fiat currency (such as USD) directly. Different payment processors offer different solutions to this problem.

Services such as Mycryptocheckout or CoinPayments or Anypay offer the ability to diversify your payments into a range of different cryptocurrencies, but require you to set up automatic withdrawals to an *exchange* for conversion to fiat currency. Note that these listing are not endorsements, and you must complete your own due diligence and/or seek advice from a tax and investment specialist before investing.

1.9.4 Legal considerations

Tax, legal and regulatory considerations may be applicable in some jurisdictions.

1.10 Technical Guides

1.10.1 SmartCash Wallet Integration

SmartCash NodeClient is a fork of Bitcoin and the majority of functionality included in the SmartCash NodeClient Daemon can be integrated in a similar manner. Key differences relate to customizations to existing JSON-RPC commands to support unique functionalities such as InstantPay. These differences, as well as more general information, are summarized below.

- 1. General Information: SmartCash is a "Proof of Work" blockchain with attributes similar to that of Bitcoin.
 - a. Block Time: ~55 seconds per Block
 - b. Blockchain Confirmations: 6 Confirmations (or 2 in the case of InstantPay)
 - c. Github Source: https://github.com/SmartCash/Core-Smart
 - d. Release Link: https://github.com/SmartCash/Core-Smart/releases/
- 2. **JSON-RPC Interface:** The majority of Bitcoin JSON-RPC commands are unchanged making integration into existing systems relatively straightforward. For a complete listing of RPC commands see the command-line options.

It's worth noting that several key Transaction-related JSON-RPC commands have been modified to support InstantPay.

- 3. Block Hashing Algorithm: SmartCash uses the "keccak" algorithm in place of SHA256 used in Bitcoin. It's important to note, however, that this only affects the hashing of the Block itself. All other internals utilize SHA256 hashes (transactions, merkle root, etc) which allows for most existing libraries to work in the Smart-Cash ecosystem.
- 4. **Supporting Libraries:** Due to the aforementioned differences in Hashing Algorithm only minor adjustments are required before using Bitcoin libraries on the SmartCash network. The most popular libraries have already been ported to SmartCash which has enabled support for most major programming languages. These resources are outlined in the *SDK Resources* section of this document.

1.10.2 Integration Notes

1.10.3 API Services

Several API services exist to facilitate quick and easy integration with the SmartCash network for services including:

- Transaction broadcasting
- Exchange rates
- · Currency conversion

• Invoice generation

API Services are typically leveraged to eliminate that requirement of running your own infrastructure to support blockchain interactions. This includes mechanisms such as:

- Forming and Broadcasting a Transaction to the network.
- Address generation using HD Wallets.
- Payment Processing using WebHooks.

There are a variety of options for supporting these methods, with the key differentiator being the pricing model included and supported features. The following list of API Providers attempts to outline these key features/differentiators and also includes a link to related documentation.

Insight

https://insight.smartcash.cc/api https://github.com/SmartCash/smart-insight-api

The open-source Insight REST API provides you with a convenient, powerful and simple way to read data from the SmartCash network and build your own services with it. A practical guide to getting started with the Insight API and Insight UI block explorer is available *here*.

- Features: Transaction Broadcast, WebSocket Notifications.
- Pricing Model: Free / Open Source
- Documentation: https://github.com/bitpay/insight-api

CoinPayments



https://www.coinpayments.net

CoinPayments is an integrated payment gateway for cryptocurrencies such as SmartCash. Shopping cart plugins are available for all popular webcarts used today. CoinPayments can help you set up a new checkout, or integrate with your pre-existing checkout.

- Features: Invoicing, Exchange Rates, WebHook Callbacks. CoinPayments holds Private Keys on their server allowing merchant to withdraw funds in Cryptocurrency or convert to Fiat.
- Integrations: aMember Pro, Arastta, Blesta, BoxBilling, Drupal, Ecwid, Hikashop, Magento, OpenCart, OS-Commerce, PrestaShop, Tomato Cart, WooCommerce, Ubercart, XCart, ZenCart
- Pricing Model: 0.5% Processing Fee (https://www.coinpayments.net/help-fees)
- Documentation: https://www.coinpayments.net/apidoc

Mycryptocheckout



https://mycryptocheckout.com/

Accept cryptocurrency payments for WordPress.

- 0% transaction fees
- Peer-to-peer payments
- Use any crypto wallet you want
- No KYC or product restrictions
- Accept 93+ coins

Anypay



https://anypay.global/

Anypay is Fast, Private, No-Fees, Multi-coin Point of Sale.

1.10.4 SDK Resources

SDKs (Software Development Kits) are used to accelerate the design and development of a product for the SmartCash Network. These resources can either be used to interface with an API provider or for the creation of standalone applications by forming transactions and/or performing various wallet services.

SmartCash Developer Guide



https://github.com/SmartCash/Core-Smart

The SmartCash Developer Guide aims to provide the information you need to understand SmartCash and start building SmartCash-based applications. To make the best use of this documentation, you may want to install the current version of SmartCash Core, either from source or from a pre-compiled executable.

• Documentation: https://github.com/SmartCash/Core-Smart

SmartCard SDK



https://github.com/SmartCash/Core-Smart

Easy create your smartcard from your website

Open with button click - Auto Open False

Open on page load - Auto Open True

```
<script type="text/javascript" src="https://card.smartcash.cc/js/v1/smartcard.js"
data-auto-open="true">
data-generate-csv="true"></script>
```

Add CSV to zip - Generate CSV True

```
<script type="text/javascript" src="https://card.smartcash.cc/js/v1/smartcard.js"
data-auto-open="true"
data-generate-csv="true"></script>
```

Python: python-smartcashrpc

https://github.com/leoreinaux/python-smartcashrpc

python-smartcashrpc AuthServiceProxy is an improved version of python-jsonrpc.

- sends Basic HTTP authentication headers.
- parses all JSON numbers that look like floats as Decimal, and serializes Decimal values to JSON-RPC connections.
- · Documentation: https://github.com/leoreinaux/python-smartcashrpc/blob/master/README.rst
- Repository: https://github.com/leoreinaux/python-smartcashrpc

PHP: Bitcoin-PHP

https://github.com/Bit-Wasp/bitcoin-php

Bitcoin-PHP is an implementation of Bitcoin with support for SmartCash using mostly pure PHP.

- Platform: PHP
- Documentation: https://github.com/Bit-Wasp/bitcoin-php/blob/master/doc/Introduction.md
- Repository: https://github.com/Bit-Wasp/bitcoin-php

Python: PyCoin

https://github.com/richardkiss/pycoin

PyCoin is an implementation of a bunch of utility routines that may be useful when dealing with Bitcoin and Smart-Cash. It has been tested with Python 2.7, 3.6 and 3.7.

- Platform: Python
- Documentation: https://pycoin.readthedocs.io/en/latest/
- Repository: https://github.com/richardkiss/pycoin

.NET: NBitcoin



https://github.com/MetacoSA/NBitcoin

NBitcoin is the most complete Bitcoin library for the .NET platform, and has been patched to include support for SmartCash. It implements all most relevant Bitcoin Improvement Proposals (BIPs) and SmartCash Improvement Proposals (DIPs). It also provides low level access to SmartCash primitives so you can easily build your application on top of it.

- Platform: .NET
- Documentation: https://programmingblockchain.gitbooks.io/programmingblockchain/content/
- Repository: https://github.com/MetacoSA/NBitcoin

BlockCypher



https://www.blockcypher.com

BlockCypher also offers client SDKs.

• Platform: Ruby, Python, Java, PHP, Go, NodeJS

• Repositories: https://www.blockcypher.com/dev/SmartCash/#blockcypher-supported-language-sdks

GoCoin

https://gocoin.com

- Platform: JavaScript, PHP, Java, Ruby, .NET, Python
- Repositories: https://gocoin.com/docs

1.10.5 InstantPay

InstantPay is a feature provided by the SmartCash network that allows for 0-confirmation transactions to be safely accepted by Merchants and other service providers. Secured by the Smartnode Network, this mechanism eliminates the risk of a "Double Spend" by locking transaction inputs for a given transaction at a protocol level.

InstantPay Transactions vs. Standard Transactions

From an integration perspective there are only minor differences between an InstantPay Transaction and a Standard Transaction. Both transaction types are formed in the same way and are signed using the same process; the key difference is the fee structure and input requirements for InstantPay.

- 1. Fee Structure: InstantPay utilizes a "per-input" fee of 0.001 SmartCash per Input.
- 2. Input Requirements: All inputs for an InstantPay transaction must have at least 2 confirmations.
- 3. Amount: Total amount InstantPay limits the total coins that can be locked in a transaction to %1 SMART and must less than 100k. You may need to use a normal transaction to split large deposits before you can send with InstantPay.

In the event that a given transaction does not meet both criteria it will revert to a standard transaction.

Receiving InstantPay Transactions

InstantPay transactions are handled in the same way as a Standard Transaction, typically through JSON-RPC, Insight API, or an internal notification script / service that is configured at a server level.

- 1. JSON-RPC: The following RPC commands will include InstantPay-related information. Within the response you'll find an "InstantLock" field the status of a given Transaction. This true/false (boolean) value will indicate whether an InstantPay has been observed.
- Insight API: Insight API can be used to detect InstantPay transactions and to push notifications to clients using WebSockets. The API can also be manually polled to retrieve Transaction information including InstantPay status.

Broadcasting InstantPay Transactions

InstantPay Transactions can be constructed and broadcast using an approach similar to Standard Transactions. Provided the InstantPay Fee Structure and Input Requirements are met, an InstantPay can be broadcast using JSON-RPC or Insight API as a Raw Transaction.

1. JSON-RPC: The "SendRawTransaction" RPC command can be utilized to broadcast a raw transaction using InstantPay. When utilizing this command be sure to set both optional parameters as "true"

sendrawtransaction "hexstring" (allowhighfees InstantPay)
sendrawtransaction "hexstring" true true

More Information: https://github.com/SmartCash/Core-Smart/

2. Insight API: Raw Transactions can also be broadcast as an InstantPay using Insight API. In this case all that is required is to POST the raw transaction using the /tx/sendix route.

More Information: https://github.com/SmartCash/Core-Smart/

Additional Resources

The following resources provide additional information about InstantPay and are intended to help provide a more complete understanding of the underlying technologies.

1.10.6 Vending Machines

1.10.7 Price Tickers

You can add a simple price ticket widget to your website using the simple code snippet generator from CoinGecko.

1.10.8 QR Codes

Many wallets are capable of generating QR codes which can be scanned to simplify entry of the SmartCash address. Printing these codes or posting the on your website makes it easy to receive payment and tips in SmartCash, both online and offline.

• In SmartCash NodeClient, go to the **Receive** tab, generate an address if necessary, and double-click it to display a QR code. Right click on the QR code and select **Save Image** to save a PNG file.

1.11 SmartHive

1.11.1 SmartHive Teams

We believe 'Core' teams are a bad idea and something that ultimately leads to inefficiency and corruption. We want to move past it and create a decentralized organizational model inspired by ant and bee colonies.

In order to create and maintain a decentralized governance structure, we are introducing two concepts SmartHive and Hive Structuring Teams (HST). SmartHive enables anyone that holds coins to vote on proposals submitted by the community. SmartHive will be the lifeblood of the project, which will allow anyone to get involved and submit proposals, helping to generate organic growth at a grassroots level, creating a bottom-up management structure.

The Hive Structuring Teams (HST) will lay the groundwork for the project and set the framework for the project's evolution. Initially, the Hive Structuring Teams will operate three separate departments in charge of vital project areas, such as development, outreach and support. Each of these three departments will end up employing people. As teams grow larger, bigger than eight members, more teams will be created, each with its own budget and team coordinator, operating autonomously without a central command and control structure in place.

The Hive Structuring Teams will consist of people directly involved in the project on a day to day basis, while the overall SmartHive will aim to enrol and engage community members and other 3rd parties interested in growing SmartCash into a global worldwide currency.

Right now we have 6 teams https://smartcash.cc/hive-teams/

1.11.2 Discussion

Are forum where peoples discussing about Pre-Proposal

https://forum.smartcash.cc/c/smarthive

1.11.3 SmartVote

All Smartcash users deserve a voice, from day #1. That is why Smartcash is pioneering in the crypto-governance space and was built around Smarthive, the platform that lets all users vote on the direction of the project. More information will be released soon describing this platform in depth.

Basically, each SmartCash coin is worth 1 vote, to keep it simple.

SmartHive Governance Portal

You can learn more about SmartCash Governance in the following sections:

Understanding SmartVote

One of the greatest challenges of building a cryptocurrency platform is ensuring you create a decentralized system of governance to manage, fund, maintain and expand the project. This key element has been absent in every major currency to date, so the natural response is to create a not-for-profit foundation that is tasked with maintaining the core protocol and promoting the coin, but is not really connected to the coin holders in any meaningful way. This approach has a few issues that have been made evident from the experience of older crypto currency platforms.

Current crypto foundations are not related to the currency itself by any mechanism that is included in the protocol and are not designed to outlive early adopters when they lose interest. The foundation then struggles to maintain funding until it implodes and core development of the protocol is left scrambling for funding or depending on charity that can't be counted on and does not allow for proper budgeting and planning. Donations are also unfair to donors because there are always free riders that benefit from the effort done by others without contributing. Other projects have financed themselves by premining coins or running prelaunch sales, which is not a great solution either because control of the funds is centralized and at that stage it is impossible to quantify the future needs of the project.

Through the network of full nodes and the collateral requirement, SmartCash already has a decentralized network of smartnode operators that are heavily invested in the future of the currency, and that as hive structuring teams can act as stewards of the core protocol development, outreach, web, quality assurance, support. We propose a decentralized management system based on the voting mechanism.

All SmartCash holders can vote on projects to be funded from the community hive budget. There is no cost or fee to vote.

Project treasury and smartvote

The system works as a decentralized voting mechanism set up in the rules governing the blockchain, where budgets for specific projects are proposed, then the SmartCash holders as a whole vote on them. Each project, if it passes, is added to the total budget and paid from the project treasury to the person doing the work.

Reward

To guarantee long term sustainability of the blockchain, the network keeps a portion of the block rewards back as new blocks are created, with the smartnode operators tasked to act as stewards and invest in the maintenance and expansion of the network. This results in faster development and promotion, creating a virtuous cycle that benefits all actors, including miners, smartnode operators, investors and users. More importantly, this gives the blockchain itself a self-preservation mechanism that is beyond the control of any individual.

Contractors and proposals

Contractors of the blockchain can be developers, outreach professionals, team leaders, attorneys or even people appointed to do specific tasks. Proposals generally begin life as simple pre-proposal forum posts on the SmartCash Forum, where feedback and suggestions are solicited from the general community. Once the proposal owner decides they have a reasonable chance of passing their proposal, he created proposal here https://vote.smartcash.cc/. A fee of 100 SmartCash is associated with this action to prevent spam and ensure only serious proposals make it to this stage. Several tools exist to allow smartcash holder to comfortably review and vote on proposals.

Proposal websites

The community has gathered around SmartHive Governance Portal as a website to facilitate discussion and voting on proposals.

https://vote.smartcash.cc/TermsAndConditions

Conclusion

We envision a future in which this model of transparent, unbreakable and verifiable contribution to the common good, in combination with continuous participation of the crowd through active voting, is utilized to manage organizations that are owned or operated jointly by its members, who share the benefits and responsibilities of those collectives, like governments, cooperative corporations, unions, DAOs, cryptocurrencies, etc. We call this model Community Governance via Blockchain.

Using SmartVote

SmartCash SmartHive is voting and funding platform. This documentation introduces and details the theory and practice to use the system.

Understanding the process

Introduction

- Smarthive consists of three components: Proposals, Votes, and Project treasury
- Anyone can submit a proposal for a small fee
- Each valid smartcash holder can vote for, against or abstain on proposals
- Approved proposals become Fund Allocated
- Fund Allocated are paid directly from Project Treasury to the proposal owner

Proposals

- Proposals are a request to receive funds
- Proposals can be submitted by anyone for a fee of 100 SMART. The proposal fee is irreversibly destroyed on submission.
- Proposals cannot be altered once submitted

Votes

- Votes are cast using the address that have >= 1 SMART
- Votes can be changed at any time

Proposal structure

The following information is required to create a proposal:

- proposal-name: a unique label
- Summary: a proposer-created webpage
- Forum Proposal Link: forum post containing detailed proposal information
- block-start: the requested start of proposal payments
- SmartCash Payment Address: the address to receive proposal payments
- Total Cost: the requested payment amount
- And More

Detail here: https://vote.smartcash.cc/Proposal/Create

Persistence

- 1. Proposal Funds can not be transferred back to the community as a kickback, rebate, or donation nor can they be used to fund third parties. Proposals that have this objectives will be removed.
- 2. Your proposal will be validated by our team before it goes live. We will check if it does not hurt any of the terms, only after this will be released on the site
- 3. Proposals will be open for 14 days. If during the last day of voting, the status change, the proposal will be extended for another 24h
- 4. After submitting, you will not be able to edit the proposal

Templates

The following two Microsoft Word templates are available to help facilitate standardized proposal submission and updates. Usage is recommended, but not required.

- Project Proposal Template
- Project Status Update Template

Creating proposals

Once you have prepared the text of your proposal and set up a website or forum post, it is time to submit your proposal to the blockchain for voting.

For information on how to create a proposal, see

https://vote.smartcash.cc/Proposal/Create

Voting on proposals

You can voting on proposals from Web wallet, NodeClient Wallet, Electrum Wallet and any kind of wallet that support sign message.

7 Steps to a Successful Proposal

https://smartcash.freshdesk.com/support/solutions/articles/35000027140

Proposals in the SmartCash governance system are subject to voting by SmartCash holder. So, like any voting, you need to convince the voters that your proposal should pass. Here are some key points to consider in every proposal:

- **Keep your proposal clear** Your proposal should have a clear title, followed by a short and simple description of the objectives. Explain early in your proposal exactly how it will benefit the SmartCash network, how much SmartCash you are requesting, how you arrived at this value, and finally who you are and how you plan to do the work. SmartCash holder should be able to immediately get an idea of what you are proposing from the first few lines of your proposal.
- **Run a pre-proposal discussion** Get feedback from the community before you post your proposal to the blockchain. A discussion period of around two weeks will help you find out if someone has proposed something similar in the past, and whether it succeeded or failed. There are pre-proposal channels on the forum consider the discussion on these channels to be the research phase of your proposal. Later, you can post a link to the forum discussion when your proposal goes live to show you are including community feedback in your work.
- Manage your identity and reputation The SmartCash community is one of the network's strongest features, and newcomers are always welcome. However, because of the way proposals work, there needs to be reasonable trust that the work promised in the proposal will be completed if it passes. If you are new, consider starting with a smaller proposal first to prove your ability to deliver on time and budget. Attaching your real name or Keybase identity to a proposal also helps build trust. If you are a making a large proposal, get a team together and nominate (or hire) one person to serve as community liaison, since posting from multiple accounts can be confusing.
- **Run an enthusiastic campaign for your proposal** Proposals with a video or website have a far greater chance of succeeding! Uploading a video gives your proposal a human touch and a chance to convey your enthusiasm in a way that isn't always possible in text. Post your video to the forum, become a regular on Discord or run a webinar to explain the proposal and answer questions. Put some work in before you ask for funding to demonstrate your involvement with SmartCash but don't be annoying and spam many channels asking for votes.
- **Demonstrate your commitment to the network** If you are asking for significant funding to start up or expand a forprofit business built on SmartCash, you need to explain why and for how long this funding is required, and what you are offering in return. It can be very helpful to show you have skin in the game by matching the contribution provided in SmartCash with funds from your own business or investors.
- Post your proposal early and make yourself available for questions To give the SmartCash holder enough time to consider, discuss and vote on your proposal, you must post it well in advance of the voting deadline. The first

few hours of discussion between SmartCash holder typically bring up a lot of questions and can be critical to influence opinion and voting, so make yourself available during this time.

- Keep the community updated when your proposal passes Your proposal should include details of how you plan to keep the community and network informed of your work. Meet your commitments and post regular reports so your output is clear, and make yourself available on social channels to answer questions. Remember, your ability to pass future proposals depends on your demonstrated ability to deliver and communicate.
- **Consider arrangements for large requests** If you are requesting a significant amount of funding, there is an understandable concern that you will deliver on your promises to the network. If your proposal is so large that uses a significant percentage of the budget, there is a risk that approving your proposal will bump smaller proposals out of the budget. Consider breaking your proposal into smaller monthly payments instead.

See this documentation for specific instructions on how to create a proposal when you are ready. Good luck!

A few additional points:

. It is not recommended to request funding for period of longer than three months. SmartCash holder don't want to see and vote on the same proposal without updates several months in a row, and price volatility makes it a risky proposition both to the network and yourself.

1.12 SmartRewards



Just keep your SMART on your wallet at least a month

In the upcoming 1.3 Phoenix release, one of the major changes will be the introduction of the SuperRewards. SmartRewards has been a way to earn passive income for all SmartCash holders. But now we have added a higher deposit tier that qualifies an address as SuperRewards. The 1.3 release will change block rewards to provide 60% allocation to SmartRewards, helping to effectively distribute incentives to community participants and receive higher rewards.

Users will need to activate each address to start getting SmartRewards or SuperRewards, but the process is simple and fast.

The yearly yield will depend on how long the users hold coins at their addresses. The gif below shows how the estimated yearly yield changes depending on the holding period (1 week, 2 weeks, 3 weeks, 4 weeks). Users will be able to get the maximum yearly yield if they hold the coins for 4 weeks or more. Additionally, users will also be able to receive SuperRewards if they have 1 million or more SMART at one address. SuperRewards will have a higher annual yield. The final rate will depend on how many addresses participate and qualify for the reward. It is worth to noticing that this will be significantly higher than the current percentage, since the rewards allocation is increased and only activated non SmartNode addresses are eligible for SmartRewards. The updated SmartRewards will allow to earn passive income every week.

Activation process

In a local wallet (Node Client), users simply need to open SmartRewards tab and then click on the "Send ActivateRewards" button to activate the address. Next to each address will be indicated whether the address is activated or not.

The whole process will take a few seconds.

Some third party wallets like, the Ellipal wallet, will offer the same convenience of an activation button. Any mobile or 3rd party wallet can activate by simply sending a "send all" or "send available" transaction back to the same address used to deposit funds.

1.12.1 SmartRewards FAQ'S

1.13 SmartNodes

SmartNodes are servers that run a SmartCash wallet and make decisions, such as locking transactions with InstantPay

This documentation focuses on understanding the services smartnodes provide to the network, but also includes guides on how to run a smartnode, using either a hosting provider or by setting up and maintaining your own hosting solution. The primary requirement to run a smartnode on the SmartCash network is 100000 SMART. This is known as the collateral, and cannot be spent without interrupting operation of the smartnode. The second requirement is the actual server running the SmartCash smartnode software.

Option 1: Hosted smartnode

Since operating your own server requires a certain level knowledge of blockchains and Linux server operating systems, several community members offer dedicated hosting solutions for a fee. Taking advantage of these services means the user only needs to provide the smartnode collateral and pay the hosting fee in order to receive payment from the block reward. See *these pages* for information on how to set up a hosted smartnode.

Option 2: Self-operated smartnode

Users with a deeper understanding (or curiosity) about the inner workings of the SmartCash network may choose to operate their own smartnode on their own host server. Several steps are required, and the user must assume responsibility for setting up, securing and maintaining both the server and collateral. See *these pages* for information on how to set up a self-hosted smartnode.

1.13.1 Understanding Smartnodes

This section of the documentation describes the principles and mechanisms of smartnodes and the services they provide to the SmartCash network specifically.

Simply put, a smartnode is a server with a full copy of the SmartCash blockchain, which guarantees a certain minimum level of performance and functionality to perform certain tasks related to block validation, as well as InstantPay, as the anonymity and instant transaction features in SmartCash are called. The smartnodes are paid for this service, using a concept known as Proof of Service. This is in addition to the Proof of Work done by miners to secure the blockchain.

Anyone can run a smartnode. The objective is to have enough decentralization to ensure that no single person controls a significant fraction of the smartnodes. However, to avoid bloating the network with unnecessary smartnodes or encouraging reckless operators, there is one condition that needs to be fulfilled: proof of ownership of 100000 SmartCash. The coins don't need to be in the smartnode, but they need to be kept in a certain way that is transparent to the entire network. If the owner moves or spends those coins, the smartnode stops working and payment ceases.

Smartnodes are paid by the network for the Smartreward, InstantPay they provide. 10% of the block reward is paid out to the smartnodes. As more smartnodes are created, the duration between payments increases. If the collateral behind a smartnode is spent, or if a smartnode stops providing services to the network for more than one hour, it is removed

from the list until normal service resumes. In this way, smartnodes are given incentive to provide efficient and reliable services to the network.

Having so many servers holding a full copy of the blockchain and working for the coin can be extremely useful. Thanks to the reward system, there is no risk of not having enough smartnodes, and the developers can rely on them quickly deploying any new decentralized feature they want to implement. This is where the true strength of SmartCash lies - an incentivized system of thousands of distributed servers working 24x7 means that SmartCash can scale more efficiently and deploy services more quickly than a blockchain run entirely by unpaid volunteers. The more smartnodes, the better and safer the SmartCash network.

The process of setting up or upgrading a smartnode is as follows:

- 1. Set up your server and operating system
- 2. Install the SmartCash software and synchronize the blockchain
- 3. Generate a smartnode gen key and enter the private key on the smartnode
- 4. Prepare a 100000 SMART transaction

Step 1 can be omitted if you have an existing server. Steps 2 and 3 require direct access to the smartnode. Steps 3 and 4 require access to a SmartCash Wallet.

Smartnodes vs. mining

SmartCash, like Bitcoin and most other cryptocurrencies, is based on a decentralized ledger of all transactions, known as a blockchain. This blockchain is secured through a consensus mechanism; in the case of both SmartCash and Bitcoin, the consensus mechanism is Proof of Work (PoW). *Miners* attempt to solve difficult problems with specialized computers, and when they solve the problem, they receive the right to add a new block to the blockchain. If all the other people running the software agree that the problem was solved correctly, the block is added to the blockchain and the miner is rewarded.

SmartCash works a little differently from Bitcoin, however, because it has a two-tier network. The second tier is powered by smartnodes (Full Nodes), which enable instant transactions (InstantPay). Because this second tier is so important, smartnodes are also rewarded when miners discover new blocks.

The smartnode system is referred to as Proof of Service (PoSe), since the smartnodes provide crucial services to the network. In fact, the entire network is overseen by the smartnodes, which have the power to reject improperly formed blocks from miners. If a miner tried to take the entire block reward for themselves or tried to run an old version of the SmartCash software, the smartnode network would orphan that block, and it would not be added to the blockchain.

In short, miners power the first tier, which is the basic sending and receiving of funds and prevention of doublespending. Smartnodes power the second tier, which provide the added features that make SmartCash different from other cryptocurrencies. Smartnodes do not mine, and mining computers cannot serve as smartnodes. Additionally, each smartnode is "secured" by 100000 SMART. Those SMART remain under the sole control of their owner at all times, and can still be freely spent. The funds are not locked in any way. However, if the funds are moved or spent, the associated smartnode will go offline and stop receiving rewards.

Payment Determination

SmartNodes are paid at a rate of 1 node every other block.

Smartnode requirements

• 100000 Smart: Arguably the hardest part. Smart can be obtained from exchanges such as EXMO, HitBTC. For the full list of places to obtain Smart *Exchanges*.
- A computer for running a local wallet to start nodes and hold funds
- A VPS server installed Ubuntu for remote node running 24hrs/day with the following specs:
 - 2GB RAM (about half used for OS and half for daemon)
 - 20GB drive (less than 2GB used for wallet, some for OS, and some for future.)
 - 1 Static IPV4 Address per node (IPV4 address that never changes) (No IPV6)

1.13.2 Hosting Services

Several SmartCash community members offer smartnode hosting services. This service can be realized securely without the customer ever giving up control of the 100000 SmartCash required for collateral. For security reasons, it is highly recommended to keep the collateral on a hardware wallet when taking advantage of a hosting service. A list of currently available smartnode hosting services is available below.

List of hosting services

Disclaimer:

- SmartCash Hive Team is not officially involved in the provision of any of these services. Use it at your own risk.
- SmartNode shared hosting (less than 100000 SMART) requires you to trust another party with your funds and is considered high risk.
- Hosted SmartNode services allow you to keep control of your private keys and you should not be asked for your private key. (SmartNode Genkey is not your private key and is required for hosting)

stackofstake

https://stackofstake.com/

pecuniaplatform

https://pecuniaplatform.io/ - Operated by:@mark | pecunia#8051 (Discord)

1.13.3 Setup

Setting up a smartnode requires a basic understanding of Linux and blockchain technology, as well as an ability to follow instructions closely. It also requires regular maintenance and careful security, particularly if you are not storing your SmartCash on a hardware wallet. There are some decisions to be made along the way, and optional extra steps to take for increased security.

Commercial *smartnode hosting services* are available if you prefer to delegate day-to-day operation of your smartnode to a professional operator. When using these hosting services, you retain full control of the 100000 SmartCash collateral and pay an agreed percentage of your reward to the operator.

Before you begin

This guide assumes you are setting up a single smartnode for the first time. If you are updating a smartnode, see *here* instead. You will need:

- 100000 SmartCash
- A wallet to store your SmartCash, preferably a hardware wallet, although SmartCash NodeClient wallet is also supported
- A Linux server, preferably a Virtual Private Server (VPS)

A list of available documentation appears below:

• All in One - Smart Node Guide

This documentation describes the commands as if they were entered in the SmartCash NodeClient GUI by opening the console from **Help > Debug console**, but the same result can be achieved on a smartnode by entering the same commands and adding the prefix smartcash-cli to each command.

Set up your VPS

A VPS, more commonly known as a cloud server, is fully functional installation of an operating system (usually Linux) operating within a virtual machine. The virtual machine allows the VPS provider to run multiple systems on one physical server, making it more efficient and much cheaper than having a single operating system running on the "bare metal" of each server. A VPS is ideal for hosting a SmartCash smartnode because they typically offer guaranteed uptime, redundancy in the case of hardware failure and a static IP address that is required to ensure you remain in the smartnode payment queue. While running a smartnode from home on a desktop computer is technically possible, it will most likely not work reliably because most ISPs allocate dynamic IP addresses to home users.

We will use Vultr hosting as an example of a VPS, although DigitalOcean, Amazon EC2, Google Cloud, Choopa and OVH are also popular choices. First create an account and add credit. Then go to the **Servers** menu item on the left and click + to add a new server. Select a location for your new server on the following screen:

1 Server Location			
All Locations America Europe	Australia Asia		
Frankfurt	Paris	United Kingdom	Amsterdam
Germany	Franco		Netherlands
Miami	Dallas	Seattle	Atlanta
United States	United States	United States	United States
Chicago	Los Angeles	New York (NJ)	Silicon Valley
United States	United States	United States	United States
Singapore	Tokyo	Sydney	
Singapore	Japan	Austrolio	

Fig. 89: Vultr server location selection screen

Select Ubuntu 16.04 x64 as the server type. We use this LTS release of Ubuntu instead of the latest version because LTS releases are supported with security updates for 5 years, instead of the usual 9 months.

2	Server Type					
	64 bit OS 32 bit OS Application	Upload ISO ISO Library Backup	Snapshot			
	CentOS	CoreOS	O Debian	Fedora		
	Select Version	Stable x64	Select Version	Select Version		
	FreeBSD	OpenBSD	Ubuntu	Windows		
	Select Version	6 x64	1604 x84	Select Version		

Fig. 90: Vultr server type selection screen

Select a server size offering at least 2GB of memory.

Server Hostna	me & Label		
Enter server hostname SmartNode		Enter server label SmartNode	
Servers Oty:	Summary:		Danlau Neur
- 1 +	\$5.00/mo (\$0.007/hr)		Deploy Now

Fig. 91: Vultr server size selection screen

Enter a hostname and label for your server. In this example we will use SmartNode as the hostname.

Server Hostna	me & Label		
Enter server hostname SmartNode		Enter server label SmartNode	
Servers Qty:	Summary:		
- 1 +	\$5.00/mo (\$0.007/hr)		Deploy Now

Fig. 92: Vultr server hostname & label selection screen

Vultr will now install your server. This process may take a few minutes.

Click Manage when installation is complete and take note of the IP address, username and password.

SmartNode 1024 MB Server - 2073482131 Dallas --- • Running Manage

Fig. 93: Vultr server management screen

Set up your operating system

We will begin by connecting to your newly provisioned server. On Windows, we will first download an app called PuTTY to connect to the server. Go to the PuTTY download page and select the appropriate MSI installer for your system. On Mac or Linux you can ssh directly from the terminal - simply type ssh root@<server_ip> and enter your password when prompted.

Double-click the downloaded file to install PuTTY, then run the app from your Start menu. Enter the IP address of the server in the **Host Name** field and click **Open**. You may see a certificate warning, since this is the first time you are connecting to this server. You can safely click **Yes** to trust this server in the future.

You are now connected to your server and should see a terminal window. Begin by logging in to your server with the user root and password supplied by your hosting provider.

You should immediately change the root password and store it in a safe place for security. You can copy and paste any of the following commands by selecting them in your browser, pressing Ctrl + C, then switching to the PuTTY window and right-clicking in the window. The text will paste at the current cursor location:

passwd root

Enter and confirm a new password (preferably long and randomly generated). Next we will create a new user with the following command, replacing <username> with a username of your choice:

adduser <username>

Download PuTTY: latest release (0.69)

<u>Home</u> | <u>FAQ</u> | <u>Feedback</u> | <u>Licence</u> | <u>Updates</u> | <u>Mirrors</u> | <u>Keys</u> | <u>Links</u> | <u>Team</u> Download: <u>Stable</u> · <u>Snapshot</u> | <u>Docs</u> | <u>Changes</u> | <u>Wishlist</u>

This page contains download links for the latest released version of PuTTY. Currently this is 0.69, released on 2017-04-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a <u>permanent link to the 0.69 release</u>.

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the <u>development snapshots</u>, to see if the problem has already been fixed in those versions.

	Package files				
	You probably want (Not sure whether y	one of these. They include all the PuTT you want the 32-bit or the 64-bit version	Y utilities. ? Read the <u>FAQ entr</u>	<i>L.</i>)	
	MSI ('Windows Iı	astaller')			
	32-bit:	putty-0.69-installer.msi	(or by FTP)	(signature)	
	64-bit:	<pre>putty-64bit-0.69-installer.msi</pre>	(or by FTP)	(signature)	
Unix source archive					
	.tar.gz:	putty-0.69.tar.gz	(or by FTP)	(signature)	

Fig. 94: PuTTY download page



Fig. 95: PuTTY security alert when connecting to a new server



Fig. 96: Password challenge when connecting to your VPS for the first time

You will be prompted for a password. Enter and confirm using a new password (different to your root password) and store it in a safe place. You will also see prompts for user information, but this can be left blank. Once the user has been created, we will add them to the sudo group so they can perform commands as root:

usermod -aG sudo <username>

Now, while still as root, we will update the system from the Ubuntu package repository:

apt update apt upgrade

The system will show a list of upgradable packages. Press **Y** and **Enter** to install the packages. We will now install a firewall (and some other packages we will use later), add swap memory and reboot the server to apply any necessary kernel updates, and then login to our newly secured environment as the new user:

```
apt install ufw python virtualenv git unzip pv
```

(press Y and Enter to confirm)

```
ufw allow ssh/tcp
ufw limit ssh/tcp
ufw allow 9678/tcp
ufw logging on
ufw enable
```

(press Y and Enter to confirm)

```
fallocate -1 4G /swapfile
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
nano /etc/fstab
```

Add the following line at the end of the file (press tab to separate each word/number), then press Ctrl + X to close the editor, then Y and Enter save the file.

```
/swapfile none swap sw 0 0
```

Finally, in order to prevent brute force password hacking attacks, open the SSH configuration file to disable root login over SSH:

nano /etc/ssh/sshd_config

Locate the line that reads PermitRootLogin yes and set it to PermitRootLogin no. Directly below this, add a line which reads AllowUsers <username>, replacing <username> with the username you selected above. The press Ctrl + X to close the editor, then Y and Enter save the file.

Then reboot the server:

reboot now

PuTTY will disconnect when the server reboots.

While this setup includes basic steps to protect your server against attacks, much more can be done. In particular, authenticating with a public key instead of a username/password combination, installing fail2ban to block login brute force attacks and enabling automatic security updates is advisable. More tips are available here. However, since the smartnode does not actually store the keys to any SmartCash, these steps are considered beyond the scope of this guide.

Send the collateral

A SmartCash address with a single unspent transaction output (UTXO) of exactly 100000 SmartCash is required to operate a smartnode. Once it has been sent, various keys regarding the transaction must be extracted for later entry in a configuration file and registration transaction as proof to write the configuration to the blockchain so the smartnode can be included in the deterministic list. A smartnode can be started from a hardware wallet or the official SmartCash NodeClient wallet, although a hardware wallet is highly recommended to enhance security and protect yourself against hacking. This guide will describe the steps for SmartCash NodeClient.

Sending from SmartCash NodeClient wallet

Open SmartCash NodeClient wallet and wait for it to synchronize with the network.

Click **Help > Debug Windows** to open the console. Type the following two commands into the console to generate a legacy smartnode key

```
smartnode genkey
7REvo76RN2Z92kyU7NcC5FBtsRGTNXfnNuLGQQPrcgk1NKRBQSn
```

Take note of the legacy smartnode private key and collateral address, since we will need it later. The next step is to secure your wallet (if you have not already done so). First, encrypt the wallet by selecting **Settings > Encrypt wallet**. You should use a strong, new password that you have never used somewhere else. Take note of your password and store it somewhere safe or you will be permanently locked out of your wallet and lose access to your funds. Next, back up your wallet file by selecting **File > Backup Wallet**. Save the file to a secure location physically separate to your computer, since this will be the only way you can access our funds if anything happens to your computer. For more details on these steps, see *here*.

Now send exactly 100000 SmartCash in a single transaction to the new address you generated in the previous step. This may be sent from another wallet, or from funds already held in your current wallet. Once the transaction is complete, view the transaction in a blockchain explorer by searching for the address. You will need 15 confirmations before you can start the smartnode, but you can continue with the next step at this point already: installing SmartCash NodeClient on your VPS.

Install SmartCash NodeClient on VPS

SmartCash NodeClient is the software behind both the SmartCash NodeClient GUI wallet and SmartCash smartnodes. If not displaying a GUI, it runs as a daemon on your VPS (smartcashd), controlled by a simple command interface (smartcash-cli).

Open PuTTY or a console again and connect using the username and password you just created for your new, non-root user.

Option 1: Automated installation using Bash installer

To install SmartCash using Bash installer, enter the following commands after logging in:

```
cd ~
wget https://raw.githubusercontent.com/SmartCash/smartnode/master/install.sh
```

Then press enter.

Then type:

```
bash ./install.sh
```

Then press enter and answer each question the script ask.

Your system is now running as a standard SmartCash node, and is busy completing synchronisation with the blockchain.

Option 2: Manual installation

To manually download and install the components of your SmartCash smartnode, visit the GitHub releases page and copy the link to the latest $x86_64$ -linux-gnu version. Go back to your terminal window and enter the following command, pasting in the address to the latest version of SmartCash NodeClient by right clicking or pressing **Ctrl + V**:

Verify the integrity of your download by running sha256sum (optional)

Create a working directory for SmartCash, extract the compressed archive and copy the necessary files to the directory:

```
mkdir ~/.nodeclient
tar xfv smartcash-1.3.0-x86_64-linux-gnu.tar.gz
cp -f smartcash-1.3.0/bin/smartcashd ~/.nodeclient/
cp -f smartcash-1.3.0/bin/smartcash-cli ~/.nodeclient/
```

Create a configuration file using the following command:

```
nano ~/.smartcash/smartcash.conf
```

An editor window will appear. We now need to create a configuration file specifying several variables. Copy and paste the following text to get started, then replace the variables specific to your configuration as follows:

Replace the fields marked with XXXXXX as follows:

- rpcuser: enter any string of numbers or letters, no special characters allowed
- rpcpassword: enter any string of numbers or letters, no special characters allowed
- smartnodeprivkey: this is the legacy smartnode private key you generated in the previous step
- externalip: this is the IP address of your VPS

The result should look something like this:

rpcuser=VPSUSERNAME
rpcpassword=VPSPASSWORD
port=9678
daemon=1
listen=1
server=1
smartnode=1
txindex=1
smartnodeprivkev=7REvo76RN2Z92kvU7NcC5FBtsRGTNXfnNuLG00Prcgk1NKRBOSn

Fig. 97: Entering smartnode genkey data in smartcash.conf

Press Ctrl + X to close the editor and Y and Enter save the file. You can now start running SmartCash on the smartnode to begin synchronization with the blockchain:

~/.nodeclient/smartcashd

You will see a message reading SmartCash server starting.

You will see a message reading **smartcashd not synced with network! Awaiting full sync before running Sentinel.** We now need to wait for 15 confirmations of the collateral transaction to complete, and wait for the blockchain to finish synchronizing on the smartnode. You can use the following commands to monitor progress:

~/.nodeclient/smartcash-cli snsync status

When synchronisation is complete, you should see the following response:

```
{
"AssetID": 999,
"AssetName": "SMARTNODE_SYNC_FINISHED",
```

(continues on next page)

(continued from previous page)

```
"Attempt": 0,
"IsBlockchainSynced": true,
"IsSmartNodeSyncStarted": true,
"IsSmartnodeListSynced": true,
"IsWinnersListSynced": true,
"IsSynced": true,
"IsFailed": false
```

Continue with the next step to construct the ProTx transaction required to enable your smartnode.

Start your smartnode

Start from SmartCash NodeClient wallet

Identify the transaction

If you used an address in SmartCash NodeClient wallet for your collateral transaction, you now need to find the txid of the transaction. Click **Help > Debug Windows** and enter the following command:

smartnode outputs

{

}

This should return a string of characters similar to this:

"la2f7cf157cc8f0bfc35562b69039b05197cd909e6b41a1730bcacd205fe984e " : "1",

The first long string is your transaction hash, while the last number is the index.

Add the private key to your smartnode configuration

edit smartnode.conf

the content will look like this:

```
smartnode1 46.77.121.22:9678_

→7QR3UF8WXgATf2ciiKqjDzF3SGFmf4e3jrxpUct4doWtyfetmyM_

→1a2f7cf157cc8f0bfc35562b69039b05197cd909e6b41a1730bcacd205fe984e 1
```

Double check ip of your VPS and the info in the smartnode.conf in Desktop wallet (when have as many VPS will have as many row in smartnode.conf)

Warning: Wait at least 15 confirmations (if you attempt to start your smartnode before you get 15 confirmations, your smartnode won't work and you have to restart procedure). You can check number of Collateral confirmations in explorer.

Go to SmartNodes tab > My SmartNodes

and $Press\ \mbox{Start}\ \ \mbox{MISSING}$

Status column will show PRE-ENABLE and after (nearly 10 minutes) it become ENABLE

Check status in VPS

run this command in console:

smartcash-cli smartnode status

When you see this mean your smartnode are successful started:

```
"status": "Smartnode successfully started"
```

Congratulations! Your smartnode is now running.

1.13.4 Maintenance

smartnodes require regular maintenance to ensure you do not drop off the payment queue. This includes promptly installing updates to SmartCash, as well as maintaining the security and performance of the server.

Smartnode Software Update

The SmartCash NodeClient software requires regular updates in order to remain consistent with the current network consensus. Depending on whether you installed SmartCash manually or using ppa, you must follow the procedure appropriate for your smartnode, as described below.

Option 1: Updating from ppa

To update SmartCash using ppa, log in to your server and enter the following commands:

```
sudo apt-get update && apt-get upgrade -y &&
smartcash-cli stop && sleep 20 &&
smartcashd
```

Check the status of your smartnode:

smartcash-cli smartnode status

The SmartCash software on the smartnode is now updated.

Option 2: Manual update

To update SmartCash manually, log in to your server using ssh or PuTTY. If your crontab contains an entry to automatically restart smartcashd, invoke crontab -e and comment out the appropriate line by adding the # character. It should look something like this:

* * * * pidof smartcashd || smartcashd

Then stop SmartCash running:

smartcash-cli stop

Visit the GitHub releases page and copy the link to the latest $x86_64$ -linux-gnu version. Go back to your terminal window and enter the following command, pasting in the address to the latest version of SmartCash Core by right clicking or pressing **Ctrl** + **V**:

```
cd /tmp
wqet https://github.com/SmartCash/Core-Smart/releases/download/v1.3.0/smartcash-1.3.0-
```

```
→x86_64-linux-gnu.tar.gz
```

Verify the integrity of your download by running the following command and comparing the output against the value for SHA256 Checksum:

sha256sum smartcash-1.3.0-x86_64-linux-gnu.tar.gz

Extract the compressed archive and run wallet from this directory:

```
tar xfv smartcash-1.3.0-x86_64-linux-gnu.tar.gz
cd smartcash-1.3.0/bin/
smartcashd
```

You will see a message reading "SmartCash Wallet starting". Finally, uncomment the line to automatically restart SmartCash in your crontab by invoking crontab –e again and deleting the # character.

The SmartCash software on the smartnode is now updated.

Monitoring

Goto http://discord.smartcash.cc use @SmartNodeMonitor#5083

You can use this bot to monitor your SmartNodes and subscribe event notifications. Once you added your nodes with add and enabled the desired notifications with the commands below you will receive a message from the bot on each occured event!

Common commands

help - Print this help info - Print the current status of the SmartNode network faq - Print the frequently asked question

User commands

status :**enabled** - Set :enabled to 0 to disable or to 1 to receive a notification when one of your node's status changed. reward :**enabled** - Set :enabled to 0 to disable or to 1 to receive a notification each time one of your nodes received a reward. network :**enabled** - Set :enabled to 0 to disable or to 1 to enable network notifications

me - List your user info and notification states

Node configuration

add **ip0;name0 ip1;name1 ... ipN;nameN** - Add one or multiple nodes. Use a list of ip;name pairs as arguments. Example: add 23.123.213.11;Node1 22.122.212.12;Node2 update :**ip :newname** - Change the name of a node with its IP-Address remove :**ip** - Remove one of your nodes with its IP-Address

Node review

lookup **ip0** .. **ipN** - Check the payout requirements of one or multiple nodes balance - List the SMART balances of your SmartNodes detail - List all details of your SmartNodes nodes - List only the status and last payments of your nodes top :**filter** - List only the eligible nodes in the payout zone (Top 10% of the queue) per default. You can use the optional :**filter** argument to list other ranges (10 - 100). **Examples**: top 20 For the top 20% of the queue, top 40 For the top 40%. history - List information about past payouts of your SmartNodes

Smartnode monitoring tools

Several sites operated by community members are available to monitor key information and statistics relating to the smartnode network.

Block Explorers

Since SmartCash is a public blockchain, it is possible to use block explorers to view the balances of any SmartCash public address, as well as examine the transactions entered in any given block. Each unique transaction is also searchable by its txid. A number of block explorers are available for the SmartCash network.

- CoinCheckup offers a range of statistics and data on most blockchains, including SmartCash.
- explorer.smartcash.cc and insight.smartcash.cc.
- Ccore offers a SmartCash blockchain explorer and masternodes.online provide a function to view and map SmartCash smartnodes.

SmartCash smartstats smartnode

https://stats.smartcash.cc/smartnodes

This site, offers a visual representation of many key statistics of the SmartCash smartnode network, including graphs of the total smartnode count over time, price information and network distribution.

1.13.5 Advanced Topics

Installing SmartCash on Windows VPS

Install wallet as normal. The configuration file is a list of setting=value pairs, one per line, with optional comments starting with the # character.

The configuration file is not automatically created; you can create it using your favorite plain-text editor. By default, smartcash-qt will look for a file named smartcash.conf in the smartcash data directory, but both the data directory and the configuration file path may be changed using the -datadir and -conf command-line arguments.

Plat-	Path	Typical path to	configuration file		
form	to data				
	folder				
Win-	%APP-	(Vista-10) C:\Users\username\AppData\Roaming\SmartCash\smartcash.conf			
dows	DATA%	(2000-XP)	C:\Documents	and	Settings\username\Application
		Data\SmartCasl	n\smartcash.conf		

Copy and paste the following text to get started, then replace the variables specific to your configuration as follows:

(continues on next page)

(continued from previous page)

Replace the fields marked with XXXXXX as follows:

- rpcuser: enter any string of numbers or letters, no special characters allowed
- rpcpassword: enter any string of numbers or letters, no special characters allowed
- smartnodeprivkey: this is the legacy smartnode private key you generated in the previous step
- externalip: this is the IP address of your VPS

The result should look something like this:

rpcuser=VPSUSERNAME	
rpcpassword=VPSPASSWORD	
port=9678	
daemon=1	
listen=1	
server=1	
smartnode=1	
txindex=1	
<pre>smartnodeprivkey=7REvo76RN2Z92kyU7NcC5FBtsRGTNXfnNuLGQQF</pre>	rcgk1NKRBQSn

Fig. 98: Entering smartnode genkey data in smartcash.conf

1.14 Mining

Mining in the context of cryptocurrency such as smartcash refers to the process of searching for solutions to cryptographically difficult problems as a method of securing blocks on the blockchain. The process of mining creates new currency tokens as a reward to the miner. Mining is possible on a range of hardware. smartcash implements an algorithm known as *keccak*, which the miner must solve in order to earn rewards.

The simplest and most general hardware available for mining is the general purpose CPU present in every computer. A CPU is designed to be versatile but offers less efficiency than a GPU, which is designed to rapidly calculate millions of vectors in parallel. While specific CPU instruction enhancements related to cryptography such as AES or AVX can provide a decent boost, GPUs offer a significant performance increase due to their multiple pipelines capable of processing the predictably repetitive calculations associated with cryptocurrency mining. Finally, ASICs are relatively inflexible and can only process the specific function(s) for which they were designed, but at an even faster rate than the more general purpose GPUs and CPUs. The result is a currency which is more secure against brute force attacks on the smartcash blockchain.

The profitability of mining is determined by the hashrate of your mining device, the current network difficulty and the costs of your hardware and electricity. The following links provide up to date information:

- Hashrate
- Mining difficulty
- Profitability calculation tool

1.14.1 SmartNodes vs. Mining

smartcash, like Bitcoin and most other cryptocurrencies, is based on a decentralized ledger of all transactions, known as a blockchain. This blockchain is secured through a consensus mechanism; in the case of both smartcash and Bitcoin, the consensus mechanism is Proof of Work (PoW). Miners attempt to solve difficult problems with specialized computers, and when they solve the problem, they receive the right to add a new block to the blockchain. If all the other people running the software agree that the problem was solved correctly, the block is added to the blockchain and the miner is rewarded.

smartcash works a little differently from Bitcoin, however, because it has a two-tier network. The second tier is powered by *smartnodes* (Full Nodes), which enable instant transactions (InstantPay). Because this second tier is so important, smartnodes are also rewarded when miners discover new blocks.

The smartnode system is referred to as Proof of Service (PoSe), since the smartnodes provide crucial services to the network. In fact, the entire network is overseen by the smartnodes, which have the power to reject improperly formed blocks from miners. If a miner tried to take the entire block reward for themselves or tried to run an old version of the smartcash software, the smartnode network would orphan that block, and it would not be added to the blockchain.

In short, miners power the first tier, which is the basic sending and receiving of funds and prevention of doublespending. smartnodes power the second tier, which provide the added features that make smartcash different from other cryptocurrencies. smartnodes do not mine, and mining computers cannot serve as smartnodes. Additionally, each smartnode is "secured" by 100000 smartcash. Those smartcash remain under the sole control of their owner at all times, and can still be freely spent. The funds are not locked in any way. However, if the funds are moved or spent, the associated smartnode will go offline and stop receiving rewards.

1.14.2 Mining Pools

Mining smartcash in pools is more likely to generate rewards than solo mining directly on the blockchain.

Official Pool 0%Fee

- https://eu-mine.smartcash.cc
- https://asia-mine.smartcash.cc
- https://us-mine.smartcash.cc

Other pools are also available and may be advantageous for different reasons such as ping latency, uptime, fee, users, etc.:

- https://nlpool.nl/ (keccak)
- https://blake2s.com/ (keccak)
- http://antminepool.com/ (keccak)

DISCLAIMER: This list is provided for informational purposes only. Services listed here have not been evaluated or endorsed by the smartcash developers and no guarantees are made as to the accuracy of this information. Please exercise discretion when using third-party services. If you'd like to be added to this list please reach out to hoang-ton@smartcash.cc

In addition to joining a pool, you will also need to create a smartcash address to receive your payout. To do this in smartcash NodeClient wallet, see *here*.

SmartCash Mining Pool Setup

You need filled this SmartMine Signing Key Request Form https://docs.google.com/forms/d/e/ 1FAIpQLSdvllk5adChGzLBUkCHS0EpAPRtQ85T2-HXkGlerNFr6kf-gw/formResponse

Setting up the host server

Download a copy of Ubuntu Server LTS from https://www.ubuntu.com/download/server and install it on your system according to the steps described here. If you are using a VPS such as Vultr or AWS, your provider will most likely provide an option to install this system during provisioning. Ensure you enable OpenSSH server during setup so you can control your server from a remote console. Once you have access to your server, create a new non-root user if you have not already done so using the following command, replacing <username> with a username of your choice:

adduser <username>

You will be prompted for a password. Enter and confirm using a new password (different to your root password) and store it in a safe place. You will also see prompts for user information, but this can be left blank. Once the user has been created, we will add them to the sudo group so they can perform commands as root:

usermod -aG sudo <username>

Reboot your server and log in as the new user. At this point it is recommended to connect remotely using PuTTY (for Windows) or ssh (for Linux and macOS) if you have not already done so.

Setting up port forwarding

If you are on a private network behind a router, you will need to set up port forwarding for at least port 9678 (UDP/TCP). How this is done depends on your particular network router and is therefore beyond the scope of this documentation. Guides to setting up port forwarding can be found here and here.

Take note of your IP address either from your router management interface or by visiting https://www.whatismyip.com

Option 1: Automated script setup

Option 2: Manual setup

First update your operating system as follows:

sudo apt update sudo apt upgrade

Setting up smartcashd

Pool requires a full SmartCash node to be running to get block and transaction data. To download and install SmartCash, visit https://smartcash.cc/wallets/#nodeclient on your computer to find the link to the latest SmartCash Node Client wallet. Click Linux, then right-click on Download TGZ for SmartCash Node Client Linux 64 Bit and select Copy link address. Go back to your terminal window and enter the following command, pasting in the address to the latest version of SmartCash Core by right clicking or pressing Ctrl + V:

```
cd ~
```

Verify the integrity of your download by running the following command and comparing the output against the value on https://github.com/SmartCash/Core-Smart/releases/

sha256sum smartcash-1.3.0-x86_64-linux-gnu.tar.gz

Create a working directory for SmartCash, extract the compressed archive, copy the necessary files to the directory and set them as executable:

```
mkdir .nodeclient
tar xfvz smartcash-1.3.0-x86_64-linux-gnu.tar.gz
cp smartcash-1.3.0/bin/smartcashd .nodeclient/
cp smartcash-1.3.0/bin/smartcash-cli .nodeclient/
chmod 777 .nodeclient/smartcash*
```

Clean up unneeded files:

```
rm smartcash-1.3.0-x86_64-linux-gnu.tar.gz
rm -r smartcash-1.3.0
```

Create a configuration file using the following command:

nano ~/.nodeclient/smartcash.conf

An editor window will appear. We now need to create a configuration file specifying several variables. Copy and paste the following text to get started, then replace the variables specific to your configuration as follows:

Replace the fields marked with XXXXXX as follows:

- rpcuser: enter any string of numbers or letters, no special characters allowed
- rpcpassword: enter any string of numbers or letters, no special characters allowed

Press Ctrl + X to close the editor and Y and Enter save the file. You can now start running SmartCash on the masternode to begin synchronization with the blockchain:

~/.nodeclient/smartcashd

You will see a message reading **SmartCash Node Client server starting**. You can continue with the following steps and check synchronization periodically using the following command. Synchronization is complete when the number of blocks is equal to the current number of blocks in the SmartCash blockchain, as can be seen from any synchronized SmartCash wallet or block explorer:

~/.nodeclient/smartcash-cli getblockcount

Setting up Pool

CPU Mining

Please note that the prevalence of GPU and ASIC miners mean that unless you have free electricity, this is highly unlikely to be profitable!

GPU Mining

https://smartcash.freshdesk.com/support/solutions/articles/35000041431-tutorial-how-to-mine-smartcash-in-mining-pool

Tips

Reduce the number of threads for added desktop usability and heat reduction. If the CPU temperature is too high, consider fitting a new fan and check that the heat sink thermal paste on the CPU is adequate. Tweak the processor clock speed for added performance using a motherboard controller like AI Suite for Asus motherboards. Reduction of CPU core voltage will result in lower temperature but increased instability.

Try to select a pool that is nearby to reduce network latency. If the node appears slow, switch to another location. Please distribute the hashing power globally to different pools to avoid forking.

1.15 Developers

This guide can be leveraged to quickly and efficiently integrate external applications with the SmartCash ecosystem. Anyone can contribute to the guide by submitting an issue or pull request on GitHub. The documentation is available at: https://github.com/SmartCash/Core-Smart

The remaining sections available below describe practical steps to carry out common development tasks in SmartCash.

1.15.1 Translating SmartCash

Translations of all SmartCash products are managed courtesy of Transifex, which offers its own detailed documentation of all functions and features. Within Transifex, SmartCash maintains an organization which contains multiple projects and one team of translators assigned to all of the projects. Each project is assigned with one or more target languages for translation by the project maintainer. When a translator joins the team, they are able to choose the languages they feel able to translate. They can then work on any projects specifying this language as a target language.

- Transifex
- Transifex Documentation
- SmartCash translation projects
- SmartCash translators team

In general, languages with minimal regional variation are to be translated into the common language (rather than regional) target. Portuguese, for example, is simply translated into the pt target language, rather than two separate target languages pt_BR and pt_PT , for Portuguese as spoken in Brazil and Portugal, respectively. As SmartCash grows, these languages may be separated out into their regional variants by proofreaders, depending on demand. Exceptions to this rule apply where the same spoken language is written differently, such as zh_CN and zh_TW for Simplified Chinese and Traditional Chinese.

Keeping translations consistent over time as multiple translators work on each target language is a very important part of delivering a quality user experience. For this reason, if you come across any SmartCash-specific terminology such as *smartnodes*, you should use the **Concordance** search function to see how the term has been translated in the past. Transifex will also provide **Suggestions** and **History** if it recognizes a similar string in the database of past translations. Stay consistent with past language use, but also ensure your terminology is up to date with current use!



\mathbf{Q} Concordance

The following documentation describes the various projects and any special features specific to the programming language in which the product is written.

SmartCash Docs

https://www.transifex.com/SmartCash/smart-docs

This project contains all content from the SmartCash Documentation hosted at https://smartcash.readthedocs.io. Each .html page in the documentation appears as a file in the resources section, named according to the navigation steps required to open the page. The SmartCash Documentation is written in a documentation language called reStructuredText and built using the open-source Sphinx Documentation Generator. To simplify layout, most of the text has no markup or code marks at all, but hyperlinks and certain formatting must be reproduced in the target language as follows:

Inline literals Source: Type "./smartcash-qt" to run the file.

Target: Escriba "./smartcash-qt" para correr el archivo.

Note that two backticks `` before and after a word or phrase will cause that text to appear as an inline literal. This is commonly used to highlight code or commands to be typed by the user.

Bold and italic Source: To encrypt your wallet, click **Settings** > **Encrypt**
wallet.

Target: Para encriptar su billetera, haga click en **Settings** >
Encrypt billetera.

A single \star before and after a word or phrase will render it in an *italic* font, while a double $\star\star$ will render it in **bold**.

Internal hyperlinks Source: See :ref:`here <sporks>` for a brief introduction to sporks.

Target: Ver :ref:`aquí <sporks>` para una breve introducción a sporks

An internal hyperlink consists of the phrase :ref:, followed by a single backtick `, followed by some text which must be translated, followed by angle brackets with the link target < >, followed by another backtick `. Translate the text, but do not translate the text inside the angle brackets.

External hyperlinks Source: The `official SmartCash website <https://smartcash.cc>`_ also provides a list of major exchanges offering SmartCash.

Target: El `sitio web oficial de SmartCash <https://smartcash.cc>`_ también proporciona una lista de las principales Casas de cambio o Exchanges que ofrecen SmartCash.

An external hyperlink consists of a single backtick $\hat{}$, followed by some text which must be translated, followed by angle brackets with the link target $\langle \rangle$, followed by another backtick and a single or double underscore: $\hat{}$ _____ or $\hat{}$ _____. Translate the text, but do not translate the hyperlink (unless you want to link to a version of the page in the target language).

SmartCash Graphics

SmartCash visual products such as infographics, flyers and conference handouts are produced using Adobe InDesign, Adobe Illustrator or Microsoft Word and are available for use in the *Marketing section* of the SmartCash Documentation. It is important to view the finished English layout during translation in order to understand the context of the

text you are translating. For example, many words should be translated differently depending if they are a heading, a sentence or an item in a diagram.

This design can be translated into your language at Crowdin here If you translate SmartCash Graphics, please send an email to hoangton@smartcash.cc or @hoangton#6301 on Discord when you are finished to request layout in the visual design.

SmartCash Website

The SmartCash website at https://smartcash.cc can be translated into your language at Crowdin here. Please have the website open while you translate to correctly understand the context of the source text. Once your translation is complete, please send an email to hoangton@smartcash.cc or @hoangton#6301 on Discord to request a build of your translation onto the website.

SmartCash Web Wallet

The SmartCash Web Wallet at https://wallet.smartcash.cc is available for translation in poeditor here. Please have the website open while you translate to correctly understand the context of the source text. Once your translation is complete, please send an email to hoangton@smartcash.cc or @hoangton#6301 on Discord to request a build of your translation onto the website.

SmartCard & Business

The SmartCard & Business at https://card.smartcash.cc and https://business.smartcash.cc/ is available for translation in Crowdin here. Please have the website open while you translate to correctly understand the context of the source text. Once your translation is complete, please send an email to hoangton@smartcash.cc or @hoangton#6301 on Discord to request a build of your translation onto the website.

1.15.2 Compiling SmartCash NodeClient

While SmartCash offers stable binary builds on the website and on GitHub, many users will also be interested in building SmartCash binaries for themselves. This process has been greatly simplified with the release of SmartCash NodeClient available on GitHub to compile or cross-compile SmartCash for any platform.

The instructions to build SmartCash NodeClient 1.3.0 or older are available here on a previous version of this page.

Step 1. Install dependencies required for the SmartCash Wallet

Run this command:

Step 2. Installing BerkeleyDB4.8 from source

(Most Wallets use 4.8 whereas most repositories contain 5.3+):

```
cd ~/
wget http://download.oracle.com/berkeley-db/db-4.8.30.tar.gz
tar xvf db-4.8.30.tar.gz
cd ~/db-4.8.30/build_unix
../dist/configure --enable-cxx
make
```

this might take a while - even on RPi3:

```
sudo make install
export BDB_INCLUDE_PATH="/usr/local/BerkeleyDB.4.8/include"
export BDB_LIB_PATH="/usr/local/BerkeleyDB.4.8/lib"
sudo ln -s /usr/local/BerkeleyDB.4.8/lib/libdb-4.8.so /usr/lib/libdb-4.8.so
sudo ln -s /usr/local/BerkeleyDB.4.8/lib/libdb_cxx-4.8.so /usr/lib/libdb_cxx-4.8.so
```

Link libminiupnpc:

sudo ln -s /usr/lib/libminiupnpc.so.5 /usr/lib/libminiupnpc.so.10

Step 3. Compile smartcashd from source

If this is your first time compiling a binary be aware that the build process can take a while to complete:

```
$ cd ~
$ git clone https://github.com/SmartCash/Core-Smart.git # clone SmartCash repository_
⇔locallv
$ cd smartcash # Go to repository directory
$ git tag # Retrieve a list of tags and find the most recent stable tag.
$ git checkout 1.3.0 # Grab correct version
$ cd src # Go to source directory
$ sudo dd if=/dev/zero of=/swapfile bs=64M count=16 # extra swap partition for speed
$ sudo mkswap /swapfile
$ sudo swapon /swapfile
$ make -f makefile.unix # Build smartcashd. This can take up hours and will produce a_
→binary : smartcashd
$ strip smartcashd # Reduce file size by stripping symbols
$ sudo mv smartcashd /usr/bin/smartcashd && sudo chmod a+x /usr/bin/smartcashd # move_
→smartcashd and make it executable
$ sudo rm -r ~/smartcash/ # Remove directory with sources (optional)
$ sudo swapoff /swapfile# clean up the previously initiated swap
$ sudo rm /swapfile
```

You're done and should now have the smartcashd daemon installed. To run this binary simply start the daemon with:

smartcashd -daemon

You can check on your wallet daemon with:

```
smartcashd getinfo
```

Gitian

Gitian is the deterministic build process that is used to build the SmartCash Core executables. It provides a way to be reasonably sure that the executables are really built from the source on GitHub. It also makes sure that the same, tested

dependencies are used and statically built into the executable. Multiple developers build the source code by following a specific descriptor ("recipe"), cryptographically sign the result, and upload the resulting signature. These results are compared and only if they match, the build is accepted and uploaded to SmartCash.org.

Instructions on how to build SmartCash NodeClient 1.3.0 will appear here once the Docker build system for Gitian is available. Instructions to create deterministic builds of SmartCash NodeClient 1.3.0 or older are available here on a previous version of this page.

1.15.3 Testnet

Testnet is a fully functioning SmartCash blockchain with the one key exception that because the SmartCash on the network can be created freely, it has no value. This currency, known as tSmartCash, can be requested from a faucet to help developers test new versions of SmartCash, as well as test network operations using identical versions of the software before they are carried out on the mainnet. There are a few other key differences:

- Testnet operates on port 19678 (instead of 9678)
- · Launching SmartCash NodeClient in testnet mode shows an blue splash screen

To start SmartCash NodeClient in testnet mode, find your smartcash.conf file and enter the following line:

testnet = 1

Tools and links

The links below were collected from various community sources and may not necessarily be online or functioning at any given time.

• Testnet for Bitcoin: https://en.bitcoin.it/wiki/Testnet

Faucets

Explorers

Pools

1.15.4 Insight API Installation

The open-source SmartCash Insight REST API provides you with a convenient, powerful and simple way to read data from the SmartCash network and build your own services with it. Simple HTTP endpoints exist for all common operations on the SmartCash blockchain familiar from the Bitcore Insight API, as well as SmartCash-specific features such as InstantSend transactions, budget proposals, sporks and the masternode list. This documentation describes how to set up the SmartCash Insight API server and (optionally) the SmartCash Insight UI block explorer.

A standard installation of Ubuntu Linux 18.04 LTS will be used as an environment for the server. We assume you are running as a user with sudo permissions. First update all packages and install some tools and dependencies:

```
sudo apt update
sudo apt upgrade
sudo apt install npm build-essential libzmq3-dev
```

Download and extract the latest version of SmartCash NodeClient:

Your Insight API node will start up and begin to sync. Progress will be displayed on stdout. Once sync is complete, the API endpoints listed in the documentation will be available at:

```
https://<ip-address>:3001/insight-api/<endpoint>/
```

The Insight UI block explorer will be available at:

```
http://<ip-address:3001/insight/</pre>
```

1.15.5 Version History

Full release notes and the version history of SmartCash are available here:

• https://github.com/SmartCash/Core-Smart/releases

1.16 Marketing

This page includes downloads of various templates and designs intended for use as office stationary and presentations. For a visual overview of existing web and sticker designs, please see the following links.

- · Rendered designs, badges and stickers
- · Vector art package for designers

SmartCash uses the following color scheme to promote a consistent visual identity.

Color	RGB	HEX	HSV
Orange	254,198,13	#FEC60D	46.06° 94.88% 99.61%
Deep Orange	244,181,23	#f4b517	
Grey	251,252,252	#FBFCFC	180° 0.4% 98.82%
Black	0,0,0	#000000	0° 0% 0%

1.16.1 Design Materials

Brochures

An attractive brochure about SmartCash, designed for handing out at conferences and events. SmartCash brochure.

This design can be translated into your language at Crowdin here. For more information on translating SmartCash products on Crowdin, see *here*. Please contact hoangton@smartcash.cc once translation is complete to request layout of the completed translation.

LanguageDownload				
English	PDF_SmartCash_brochure_en-1.2.8			
Russian	PDF_SmartCash_brochure_ru_1.2			
German	PDF_SmartCash_brochure_DE_1.4			
Spanish	PDF_SmartCash_brochure_es_1.2.4			
Portugu	eseDF_SmartCash_brochure_ptbr_1.2			
Japanes	ePDF_SmartCash_brochure_ja_1.2			
Vietnam	ese F_SmartCash_brochure_vi_1.2			

Flyers

An attractive flyer about SmartCash, designed to be folded in half and placed on flat surfaces at conferences and events. This design can be translated into your language at Crowdin here. For more information on translating SmartCash products on Crowdin, see *here*. Please contact hoangton@smartcash.cc once translation is complete to request layout of the completed translation.

Table	3:	Fly	vers
Include	~.		

Languag	yeDownload
English	PDF SmartCash_OnePageFlyer_Community
English	PDF SmartCash_OnePageFlyer_Business
English	PDF SmartCash_flyer

Handouts

This handout is ideal for dual-sided printing as a handout for conferences.

This design can be translated into your language at Crowdin here. For more information on translating SmartCash products on Crowdin, see *here*. Please contact hoangton@smartcash.cc once translation is complete to request layout of the completed translation.

Table 4: Handouts

LanguageDownload	
English	PDF SmartCash_BusinessCard_Folded

Infographics

Presentations

An attractive presentation about SmartCash, designed to guide the audience through the basics of cryptocurrency and advantages of SmartCash.

LanguageDownload	
English	SmartCash PPTX
English	SmartCash PPTX
English	SmartCard PPTX

Table 5: Presentations

1.16.2 Business Templates

Document templates

Official SmartCash document templates.

Table 6:	Document	templates	
10010 01		compresses.	

Languag	LanguageDownload	
Word	DOCX	
docu-		
ment		
with		
cover		
page		
and		
para-		
graph		
styles		

Presentation templates

Official SmartCash presentation templates. We strongly recommend using predefined presentation slide layouts.

Table 7. Document template	Table 7	: Docur	nent terr	plates
----------------------------	---------	---------	-----------	--------

Languag	ageDownload	
PowerPo	Poimptx	
tem-		
plate		

Cards

High resolution cards for printing. Great for use as the back of business cards, or to hand out to explain and promote SmartCash.

Table 8: Cards

Name	Download version
Busines	s English
Card	

Fonts

Table 9: Fonts

Name	Download version
Sans	Download
Pro	
/ PT	
Sans	

1.17 Legal

1.17.1 Smartcash Deemed Not a Security Under Singapore Law

The SmartCash Community received good news today as it was announced that SmartCash tokens are unlikely to be considered a financial "security" under the Singapore Securities and Futures Act.

Additionally, SmartCash is unlikely be defined as a "share", "debenture", "collective investment scheme". This means that local securities laws, which can slow down adoption of the new payment platform, should not apply.

Obtaining legal opinion letters is a requirement to list on the top exchanges in the region in order to comply with local financial regulations. With the confirmation that SmartCash is indeed not a security the process for listing on exchanges within the jurisdiction of Singapore can continue.

In order to establish the legal basis for the SmartCash token, SmartCash Outreach had engaged the Singapore law firm, Christopher Bridges Law Corporation, to provide their legal opinion regarding the whether the SmartCash tokens are securities under the Singapore Securities and Futures.

SmartCash Outreach continues to work with other law firms and exchanges in other jurisdictions throughout the world to in order to expand the global reach of the SmartCash platform.

1.17.2 How the Law Applies to SmartCash

One of the most common questions we receive is How are smartnode operators treated under the US tax laws?

Tax Treatment

Block rewards

As many already know, block rewards are paid to smartnode operators in exchange for validating transactions on the SmartCash network. The IRS has stated unequivocally that "when a taxpayer successfully 'mines' virtual currency, the fair market value of the virtual currency as of the date of receipt is includible in gross income." To be sure, smartnode do not "mine", but the IRS considers using computer resources to validate Bitcoin transactions and maintain the public Bitcoin transaction ledger to constitute "mining". By analogy, a smartnode operator should also treat as regular income the fair market value of the block reward.

SmartCash Collateral

A SmartCash user may demonstrate to the network his or her control over 100000 SMART in order to run a smartnode. These tokens never leave the user's control. If at any point during the user's tenure as a smartnode operator, the user disposes of any or all of the 100000 SMART, the network automatically strips the user of his or her status as a smartnode. Under the US Internal Revenue Code, gain or loss is realized only on the "sale or exchange" of property. The term "sale" generally means the transfer of all right, title, and interest in the property transferred. A number of factors typically are considered to determine whether a sale has occurred, the most important being whether the benefits and burdens of ownership of the transferred property have passed from the transfer to the transferee. In SmartCash, the smartnode operator retains control of the 100000 SMART and simply demonstrates that control to the network. Therefore, the holding of the 100000 SMART for purposes of qualifying as a smartnode operator should not cause a taxable event to occur because the user has not transferred any of the benefits and burdens of ownership.

Capital Gains

Assuming that the 100000 SMART are sold, whether that SmartCash is a "capital asset" will determine the tax treatment of the sale. Stocks, bonds and other investment property for example, are generally treated as capital assets. Inventory, depreciable property, and stock in trade, though, are not. Assuming the smartnode operator held the 100000 SMART either for investment purposes or for purposes of qualifying as a smartnode operator, the IRS would likely treat gain or loss on the sale of those SmartCash tokens as capital in nature. Therefore, SmartCash held for a long enough period of time could be subject to the lower "long term capital gains" tax rate.

Legal Liability

As with cash or any other currency system, users may use SmartCash in connection with illegal activity. A common question we receive is whether smartnode operators can also be liable for criminal activity, simply by relaying transactions related to that activity. The fundamental legal requirement of *mens rea* makes criminal liability unlikely for smartnode operators.

Primary Liability

Almost all crimes require that a defendant have a defined *mens rea* at the time of an offense. *Mens rea* is a mental state like purposefulness, knowledge, recklessness or negligence. For example, to act with "purpose" is commonly understood as desiring as your "conscious object" the result of a crime. "Knowledge" is a less culpable mindset than "purpose" – acting with "knowledge" requires general awareness that your actions will bring about a particular crime. "Recklessness" requires disregard of a substantial risk. Finally, a person acts "negligently" if they should have been aware of a substantial and unjustifiable risk of a particular consequence of their actions, but were not.

Most smartnodes have no awareness, while relaying SmartCash transactions, of the identity of the users involved, the ultimate destination of users' funds, or any other circumstances of SmartCash transactions. As such, it would be difficult for a prosecutor to demonstrate that a smartnode operator who facilitated an illegal transaction merely by relaying the transaction would have a culpable *mens rea*.

Secondary Liability

Even if someone is not the principal actor in the commission of a crime, that person can be secondarily liable for their involvement in it. As such, we are sometimes asked whether smartnode operators, by their involvement in relaying SmartCash transactions, could be "aiding and abetting" or "conspiring" to commit a crime that might involve SmartCash. Generally speaking, aiding and abetting requires that the defendant (i) seek by his action to make the crime succeed and (ii) act with the same *mens rea* as required for the principal offense.

No matter the requisite *mens rea* of a particular principal offense committed by a SmartCash user, it is unlikely that a mere smartnode operator, without more, could be found to have "aided and abetted." To be sure, the smartnodes do provide assistance in the principal offense – in that smartnode action is required to process all SmartCash transactions. However, the smartnodes would not have the requisite *mens rea* to satisfy the requirements of aiding and abetting liability. smartnode operators have no readily available information about the purpose or consequences of users' SmartCash transactions, or even the originating identity of the sender of funds. As such, so long as a sufficient diversity of non-criminal transactions occur on the SmartCash network, they would not harbor even the least culpable *mens rea* (i.e. negligence) with respect to a user relaying or receiving SmartCash in furtherance of a particular crime.

"Conspiracy" liability is even less likely. Conspiracy generally requires i) an agreement to commit a crime, ii) knowledge of the unlawful purpose of the agreement, iii) intent to further the unlawful purpose, and iv) an act in furtherance of the conspiracy. None of these requirements are met by mere smartnode operators.

Exchange Liability

The Bank Secrecy Act (BSA) is the law that primarily governs exchanges in the United States. The BSA does not contain any prohibition on supporting SmartCash transactions. Indeed, the BSA take a flexible, risk-based approach to regulation and contemplate that financial institutions will enter into lines of business with new risks. This risk-based approach requires, at the outset, an independent risk assessment. By and large, the risks faced by exchanges who begin to support SmartCash will be similar to the risks associated with other virtual currencies.

• Exchanges should consider revising their risk assessments and AML policies to account for the unique characteristics of SmartCash. For example, including blockchain addresses in Suspicious Activity Reports (SARs) will be less descriptive and effective for investigations based on such addresses. Exchanges might consider adding additional context and explanation in SARs.

1.17.3 ATM & Fiat Compliance

Introduction

An aspect that required legal research is what are the compliance requirements to facilitate SmartCash-fiat exchange. This can be in the form of running ATM kiosks or using other mechanisms to personally offer SmartCash to fiat exchange services.

For this purpose we found **Christopher Bridges**, who have a lot of experience working in compliance for digital money services. They have put together a compliance program that will be shared with anyone launching a SmartCash service if they require it. This includes:

- A finCEN BSA compliant written Principal MSB/KYC AML Program designed to prevent the Principal MSB from being used to facilitate money laundering and the financing of terrorist activities
- A comprehensive risk based assessment by a third party compliance consultant expert
- Banking Secrecy Act (BSA), Anti-Money Laundering (AML), and Know Your Customer (KYC) regulations
- FinCEN requirements and guidance
- Written and on-going AML compliance programs
- State-by-state money transmitter laws and licensure requirements

We now have full documentation of the program from Cogent Law, the package includes:

- BSA AML Compliance training
- KYC CDD Policy
- AML Program
- Risk Assessment
- Surveillance and Monitoring Policy
- Different Reports to file in different situations

The documents have a format as if they were created for the fictional company "Initech" and then the interested SmartCash entrepreneur would have to replace it with their real corporation. We believe that some counsel would still be advisable for an investor launching a service but this will help provide a solid understanding of the requirements and get them 80% of the way. This should also be highly educational and help the SmartCash investor be more at ease as it enlightens a typically obscure topic for most crypto enthusiasts.

This program is directed at SmartCash investors that are interested in running a compliant service in regulated markets. If some users prefer to run services without a compliance program or don't need one, we completely respect that too.

SmartCash ATM Compliance Program

- 1. Please first read the Quick Guide To Operating a Virtual Currency Kiosk. This guide will guide you through the initial steps of registering as an MSB. It is crucial you start here.
- 2. Listen to the Audio Guides on BSA AML and read the BSA AML Training Manual
 - BSA AML Compliance Training manual
 - AML 101
 - AML In Depth
- 3. Get familiar with the documentation you will need by looking at the following templates
 - Surveillance and Monitoring
 - KYC CDD Policy
 - AML Program
 - Risk Assessment
- 4. Get familiar with some of the reports you may need to file as part of operations. In the Quick Guide in Step 1, we go through registration on the E BSA filing system. This step also provides example CTR (Currency Transaction Report) and SAR (Suspicious Activity Report) report samples. SmartCash operators should be familiar with these as they may need to file them as part of operations. Sample reports:
 - Currency Transaction Report
 - Suspicious Activity Report
- 5. If you would now like to move forward with your project, we recommend getting in contact with Christopher Bridges Law Corporation, our recommended legal counsel. Preferential rates are available for SmartCash users. Please email: seccbridges@cbridgeslaw.com