UNIVERSITY of NICOSIA

Session 5

Bitcoin Core Functionality (Optional)

Bitcoin Core Installation, Command Line Interface, Exploring Transactions

DFIN 511: Introduction to Digital Currencies

Objectives

- Explore Bitcoin Core Functionality

Goal of this Session:

- Go through the Bitcoin Core installation process
- Learn how to get information from the Bitcoin blockchain by using some basic commands via the Command Line Interface (CLI)

This is an optional section, since it is more technical and students with a business background may find it challenging to follow through. However, we encourage all to invest some extra time to go through from this session, in order to understand some 'technical' aspects of Bitcoin Core.

Agenda

1. Bitcoin Core Installation
2. Bitcoin Core Minimum Requirements
3. Command Line Interface (CLI)
4. Bitcoin Core Commands: Getting Network Information
5. Bitcoin Core Commands: Managing Your Wallet
6. Exploring Transactions

Session 5: Bitcoin in Practice 2

1. Bitcoin Core

Bitcoin Core

- Any computer that connects to the Bitcoin network is called a node. Nodes that fully verify all transactions and blocks according to the consensus rules of Bitcoin are called full nodes.
- “A full node is a program that fully validates transactions and blocks. Almost all full nodes also help the network by accepting transactions and blocks from other full nodes, validating those transactions and blocks, and then relaying them to further full nodes.”
- The most popular software implementation of full nodes is called Bitcoin Core. You can download the Bitcoin installer from https://bitcoin.org/en/download or https://bitcoincore.org/en/releases/
- It is recommended for users who wish to download and experiment with Bitcoin Core, to ensure that they have the storage and bandwidth capabilities.

Read More: https://bitcoin.org/en/full-node

Bitcoin Core Requirements

Disk space
200 GB

Download
500 MB/day (15 GB/month)*

Upload
5 GB/day (150 GB/month)

Memory (RAM)
1 GB

System
Desktop
Laptop
Some ARM chipsets >1 GHz

Operating system
Windows 7/8.x/10
Mac OS X
Linux

* Plus a one-time 195 GB download the first time you start Bitcoin Core.

Note: Disc space: 400GB for transaction indexing

Read More: https://bitcoin.org/en/full-node#what-is-a-full-node

Advantages of Running a Full Node

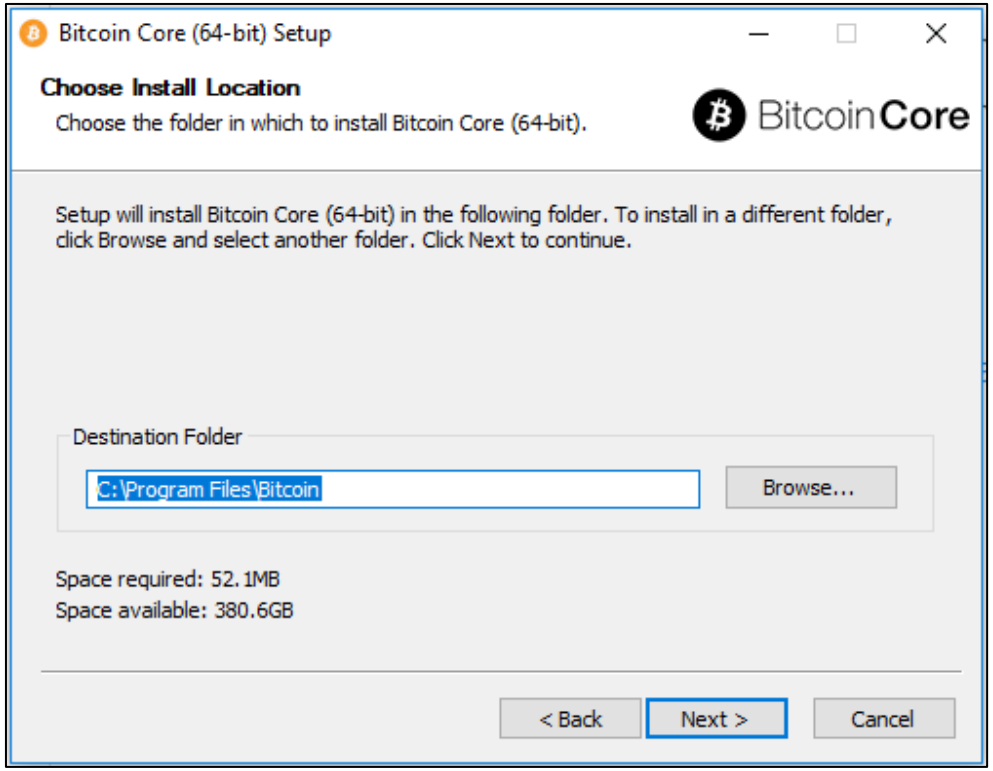
There is a number of reasons why one should consider running a full node:

- Because the full node is validating all transactions and blocks, it is much harder for someone to trick you with false transaction data, and other attacks that affect lightweight wallet users.
- In the event of a fork or other consensus event, you will have a 'voice' through the software you choose to run and how the network evolves.
- Lightweight wallets leak information about your addresses and coins when they query third-party servers for information about the blockchain. Asking your own node instead gives you a higher degree of privacy.
- Makes the network more robust. The more nodes running the Bitcoin software, the more peers there are to broadcast and validate transactions and blocks, making it harder for any single actor or group to control information flows.

Session 5: Bitcoin in Practice 2

2. Bitcoin Core Installation Process

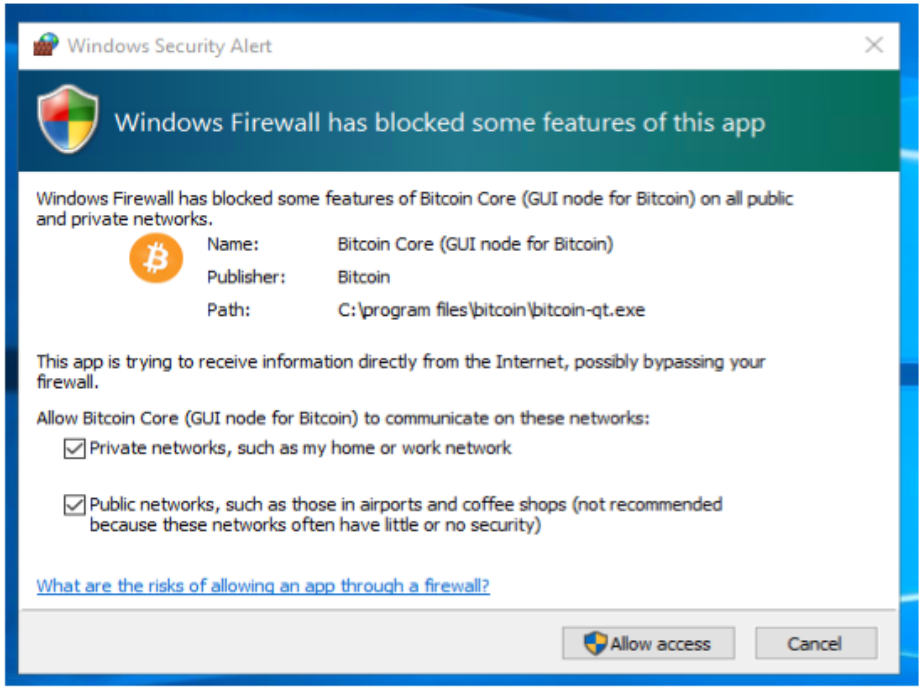
Bitcoin Core: Installation in Windows



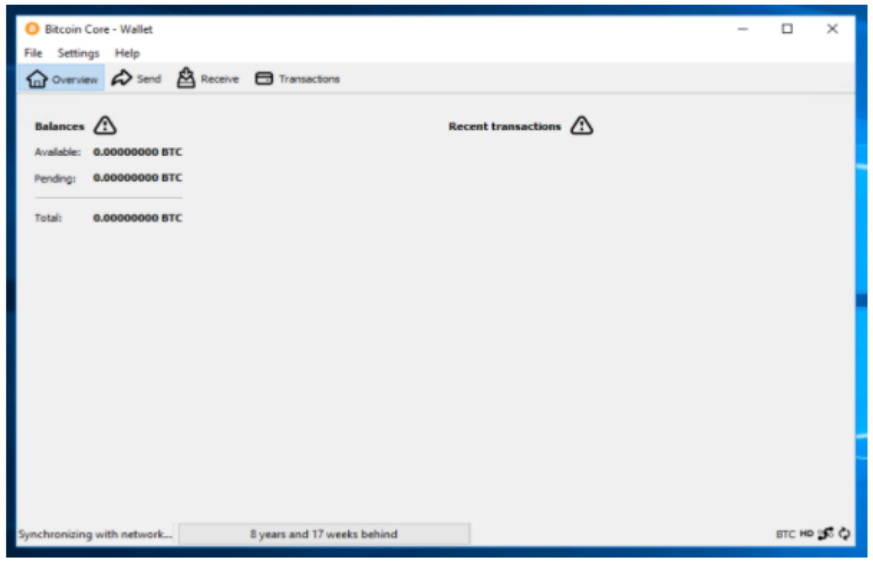
Source: https://bitcoin.org/en/full-node#windows-10

Bitcoin Core: Installation in Windows

Your firewall may block Bitcoin Core from making outbound connections. It's safe to allow Bitcoin Core to use all networks. (Note: you will still need to configure inbound connections as described later in the Network Configuration section.)



Bitcoin Core GUI will begin to download the block chain. This step will take at least several days, and it may take much more time on a slow Internet connection or with a slow computer. During the download, Bitcoin Core will use a significant part of your connection bandwidth. You can stop Bitcoin Core at any time by closing it; it will resume from the point where it stopped the next time you start it.



After download is complete, you may use Bitcoin Core as your wallet or you can just let it run to help support the Bitcoin network.

Source: https://bitcoin.org/en/full-node#windows-10

Bitcoin Clients

Continuing with the Windows demonstration, the “Bitcoin” folder in a user’s “AppData” folder in Windows is very important because it stores (among other things):

- The Bitcoin configuration file (the user is able to generate it: https://jlopp.github.io/bitcoin-core-config-generator/)
- Your Bitcoin wallet (the wallet.dat file)
- The “blocks” folder, which stores a full copy of the blockchain (current size is over 300 gigabytes)
- "Use txindex=1 to index all transactions (400GB disk needed). Transaction indexing must be turned on before initial sync or reindex=1 will need to be set to reindex the database

Bitcoin Clients

The complete reference to the Bitcoin client Application Programming Interface (API) can be found here: https://bitcoincore.org/en/doc/0.19.0/

Using the command line interface you can:

- Get information about the status of the Bitcoin network
- Manage your wallet
- Explore and decode transactions
- Explore blocks
- Create, sign, and submit transactions with unspent outputs

Session 5: Bitcoin in Practice 2

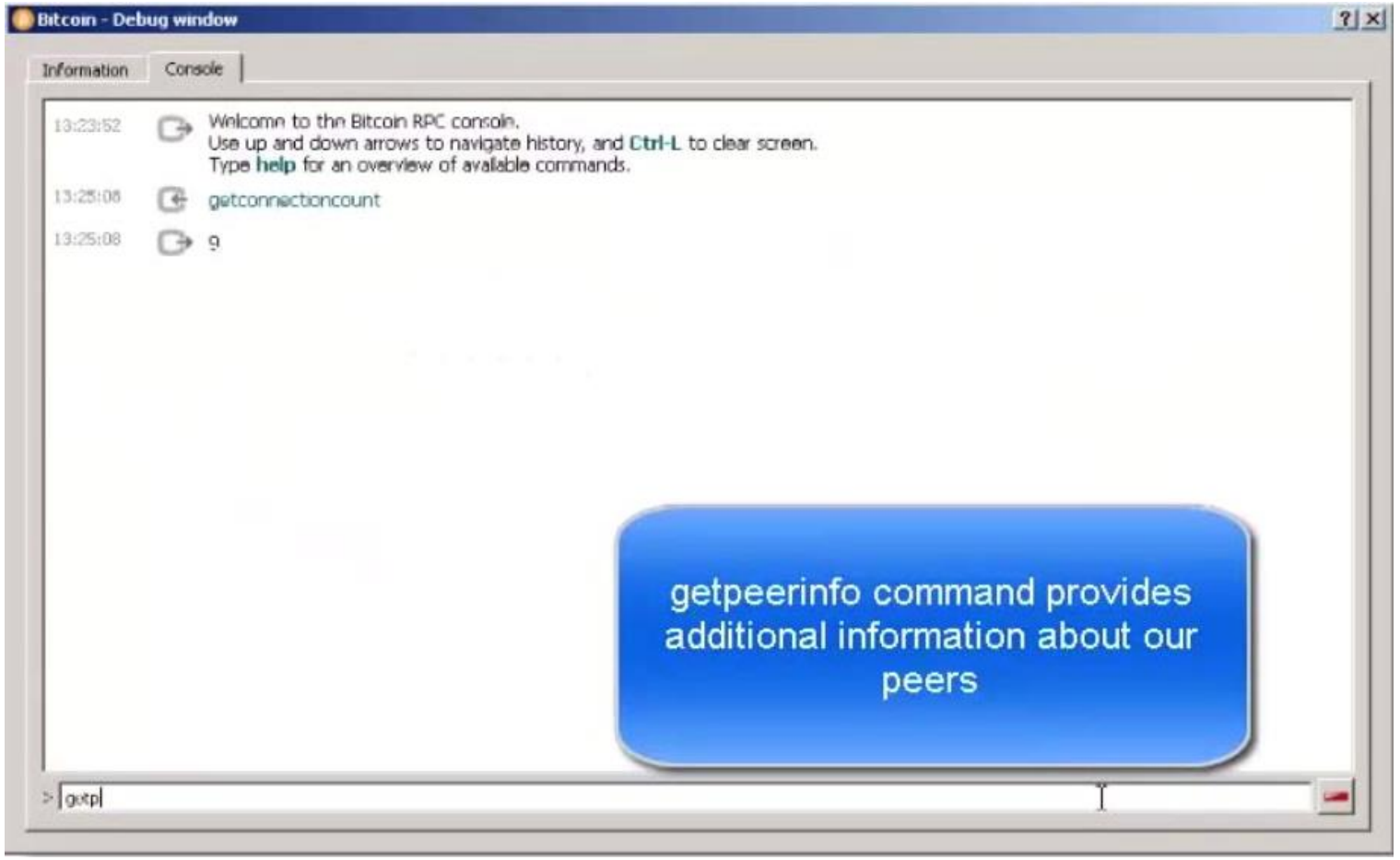
3. Basic Commands: Getting Network Information

Bitcoin Core – Getting network information

Command	Description
getconnectioncount	Returns the number of connections to other nodes.
getpeerinfo	Returns data about each connected node.
getdifficulty	Returns the proof-of-work difficulty as a multiple of the minimum difficulty.
getblockcount	Returns the height of the current fully validated chain. *(The genesis block has a height 0).*
getmininginfo	Returns an object containing information related to mining.
generatetoaddress	Mine blocks immediately to a specific address (before the RPC call returns).

Commands to use: https://chainquery.com/bitcoin-cli

Bitcoin Core – Getting network information



Video illustration with some of the previously explained commands.

Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=4vLAr6o3GPg

Session 5: Bitcoin in Practice 2

4. Basic Commands: Managing Your Wallet

Managing your Wallet

Command	Parameters	Description
getnewaddress	[account]	Returns a new bitcoin address for receiving payments. If [account] is specified, payments received with the address will be credited to [account].
dumpprivkey	<bitcoinaddress>	Reveals the private key corresponding to an address.
importprivkey	<bitcoinprivkey> [label] [rescan=true]	Adds a private key (as returned by dumpprivkey) to your wallet. This may take a while, as a rescan is done, looking for existing transactions. (*Note: There's no need to import public key, as in ECDSA, unlike RSA, this can be computed from the private key).*

Commands to use: https://chainquery.com/bitcoin-cli or https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_calls_list

Managing your Wallet

Command	Parameters	Description
getreceivedbyaddress	<bitcoinaddress> [minconf=1]	Returns the amount received by the specified address in transactions with the specified number of confirmations. It correctly handles the case where someone has sent multiple transactions to the address. Keep in mind that addresses are only ever used for receiving transactions. Works only for addresses in the local wallet, external addresses will always show 0.
listtransactions	[account] [count=10] [from=0]	Returns the most recent transactions that affect the wallet.
getaddressesbyaccount	<account>	Returns the list of addresses for the given account.
encryptwallet	<passphrase>	Encrypts the wallet with a passphrase.

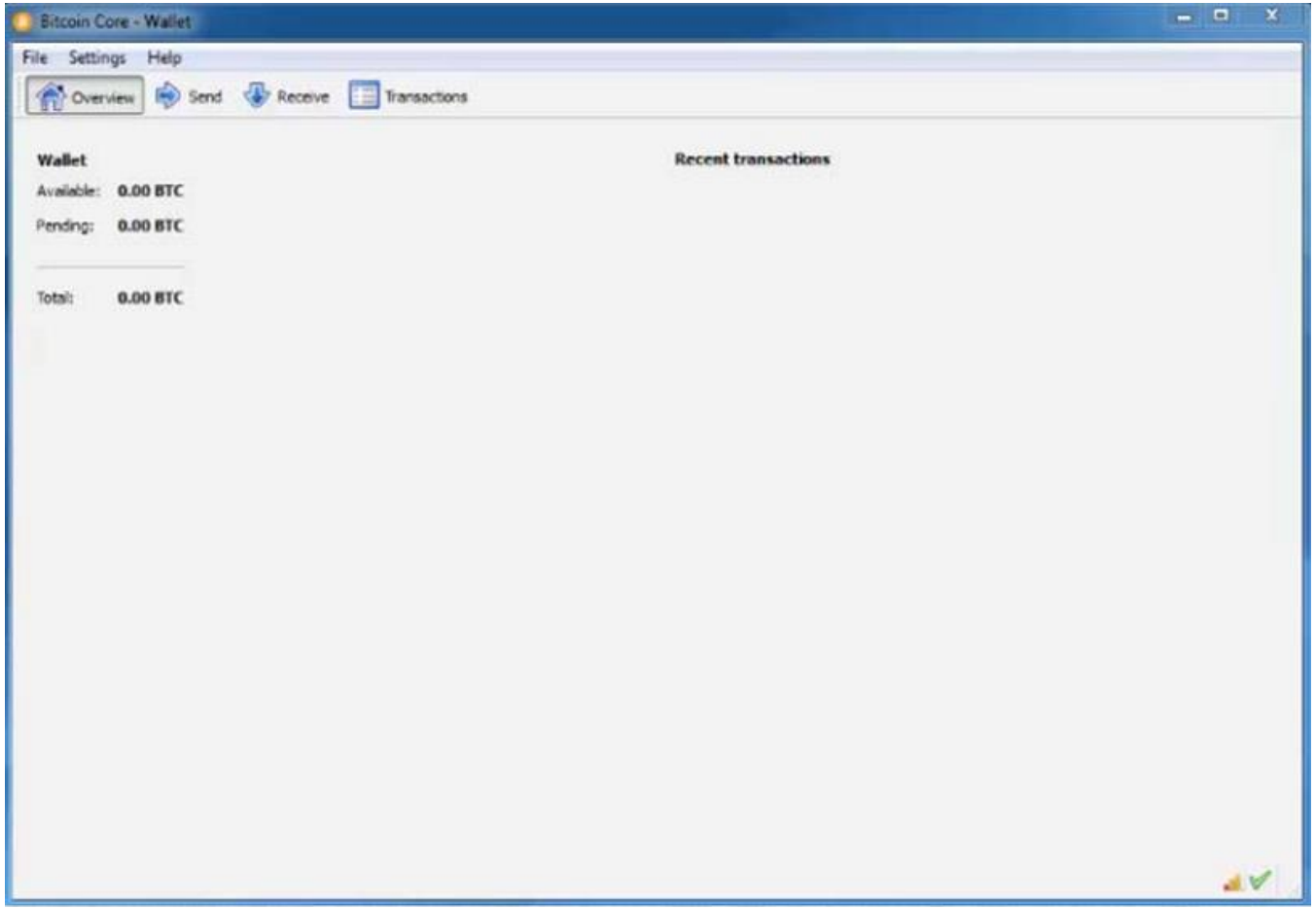
Commands to use: https://chainquery.com/bitcoin-cli

Managing your Wallet

Command	Parameters	Description
walletlock		Removes the wallet encryption key from memory, locking the wallet. After calling this method, you will need to call walletpassphrase again before being able to call any methods which require the wallet to be unlocked.
walletpassphrase	<passphrase> <timeout>	Stores the wallet decryption key in memory for indicated number of seconds.
walletpassphrasechange	<oldpassphrase> <newpassphrase>	Changes the wallet passphrase from <oldpassphrase> to <newpassphrase>.

Commands to use: https://chainquery.com/bitcoin-cli

Managing your wallet: Video illustration



Video illustration with some of the previously explained commands:

Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=kAlEAcYQuN8

Managing your Wallet

Command	Parameters	Description
gettransaction	<txid>	Gets detailed information about a transaction: • "amount" : total amount of the transaction • "confirmations" : number of confirmations for the transaction • "txid" : the transaction ID • "time" : time associated with the transaction[1]. • "details" - An array of objects containing: • "account“ "address“ "category“ "amount“ "fee"
getrawtransaction	<txid> [verbose=0]	Returns raw transaction representation for given transaction id (hex-encoded serialized transaction or a JSON object). Getrawtransaction only works if you have configured full transaction indexing (txindex=1 in the configuration file) and that requires about 380GB of disk instead of 200GB.
decoderawtransaction	<hex string>	Decodes a serialized transaction hex string into a JSON object describing the transaction.

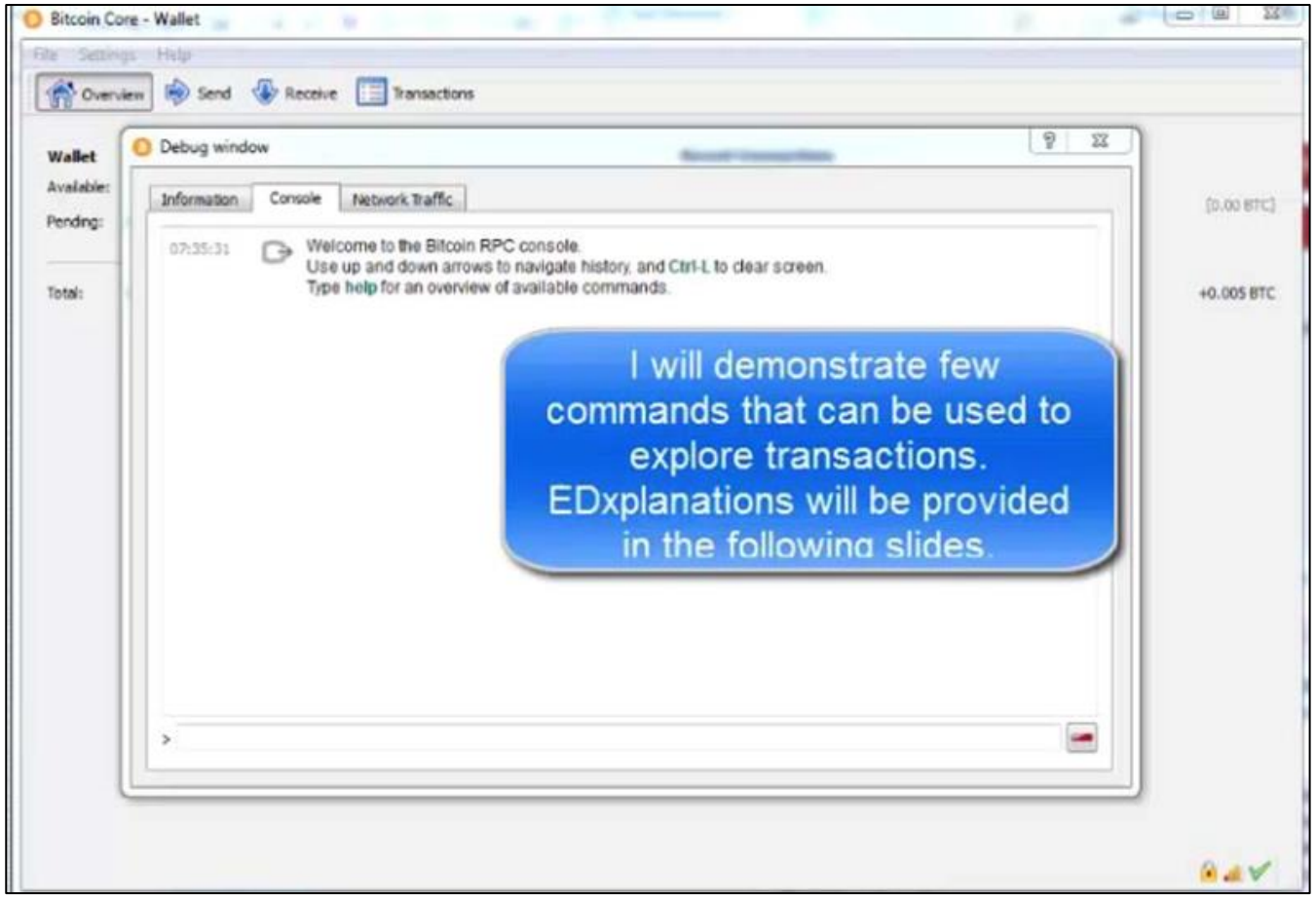
Commands to use: https://chainquery.com/bitcoin-cli

Managing your Wallet

Command	Parameters	Description
getaccountaddress	<account>	Returns the current bitcoin address for receiving payments to this account. If <account> does not exist, it will be created along with an associated new address that will be returned.
getreceivedbyaddress	<bitcoinaddress> [minconf=1]	Returns the amount received by <bitcoinaddress> in transactions with at least [minconf] confirmations. It correctly handles the case where someone has sent to the address in multiple transactions. Keep in mind that addresses are only ever used for receiving transactions. Works only for addresses in the local wallet, external addresses will always show 0.

Commands to use: https://chainquery.com/bitcoin-cli

Managing your wallet: Video illustration



Video illustration with some of the previously explained commands:

Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=IU-FDgDlCuY

Session 5: Bitcoin in Practice 2

5. Command Line Interface: Exploring Transactions

Exploring Transactions

The **gettransaction** command returns a transaction in a simplified form.

To retrieve the full transaction data, we can use two commands:

- **getrawtransaction**
- **decoderawtransaction**

The **getrawtransaction** command uses the transaction ID as a parameter and returns the full transaction as a “raw” hex string, exactly as it is on the Bitcoin network.

```
getrawtransaction
0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3
f9bdaa01c0b6c
```

```
01000000029181c1d7b6b4fc7e2f1f1ee43dfeb778468292a7b
49a3e26c9c70b268fbc9ade000000006c493046022100cc2a0d
920c154014d4e7f93878307b1b5aeab8bba25288e1f00573fe0
5cf8aac022100b8269506e5c431d55bbe4c6850e983db8b9d90
16cdece8dd7555a4ebf22816ba012102c8515f4e0512378032d
44d5ed3888bcd50be103ee26e0279f52a1fb935bb8f71ffffff
ff0e4d456390086dd622ce8be50672de7943d2a1d0ee78593a6
b4e5c7a9cb6c9c3000000008a47304402200f9e6e9bacd1f0d4
4525265455e92014faba5931a0ee8517664777d38c090d95022
000905089d5bfcf8509589984f9b79182ea1bcbf6e6ae16f765
efd8390f3c8352014104681901c41fe94cabc8e809ca1f830fd
6bc953d88254337db8ab1db9448ecd8bb2fec05f74f38abb05f
4fd5d7040f9c011365967c24672514c2a40f20dde07094fffff
fff0220a10700000000001976a9148b87c4f4c177a46de7d50b
7dd9840c16caa4728088acc07a10000000000001976a91452dad
b8a8948da05040672a11eacaecd916aa39288ac00000000
```

Note: getawtransaction requires transaction indexing txindex=1 in Core congiguration

The complete reference to the Bitcoin client Application Programming Interface (API) can be found here: https://bitcoincore.org/en/doc/0.19.0/

Exploring Transactions

The **decoderawtransaction** command shows all parts of this transaction, including the inputs and outputs.

```
decoderawtransaction 01000000029181c1d7b6b4fc7e2f1f1ee43dfeb…000

{
"txid" : "0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c",
"version" : 1,
"locktime" : 0,
"vin" : [
{
"txid" : "de9abc8f260bc7c9263e9ab4a792824678b7fe3de41e1f2f7efcb4b6d7c18191",
"vout" : 0,
"scriptSig" : {
"asm" :
"3046022100cc2a0d920c154014d4e7f93878307b1b5aeab8bba25288e1f00573fe05cf8aac02
2100b8269506e5c431d55bbe4c6850e983db8b9d9016cdece8dd7555a4ebf22816ba01
02c8515f4e0512378032d44d5ed3888bcd50be103ee26e0279f52a1fb935bb8f71",
"hex" :
"493046022100cc2a0d920c154014d4e7f93878307b1b5aeab8bba25288e1f00573fe05cf8aac
022100b8269506e5c431d55bbe4c6850e983db8b9d9016cdece8dd7555a4ebf22816ba012102c
8515f4e0512378032d44d5ed3888bcd50be103ee26e0279f52a1fb935bb8f71"
},
"sequence" : 4294967295
},
{
"txid" : "c3c9b69c7a5c4e6b3a5978eed0a1d24379de7206e58bce22d66d089063454d0e",
"vout" : 0,
"scriptSig" : {
"asm" : "304402200f9e6e9bacd1f0d44525265455e92014faba5931a0ee8517664777d38c0
```

Exploring Transactions

This transaction has two inputs as outputs of previously confirmed transactions, with IDs starting in de9a…8191 and c3c9…4d0e respectively.


```
"vin" : [
{
"txid" : "de9abc8f260bc7c9263e9ab4a792824678b7fe3de41e1f2f7efcb4b6d7c18191",
"vout" : 0,
"scriptSig" : {
"asm" :
"3046022100cc2a0d920c154014d4e7f93878307b1b5aeab8bba25288e1f00573fe05cf8aac0
22100b8269506e5c431d55bbe4c6850e983db8b9d9016cdece8dd7555a4ebf22816ba01
02c8515f4e0512378032d44d5ed3888bcd50be103ee26e0279f52a1fb935bb8f71",
"hex" : "49304…"
},
"sequence" : 4294967295
},
{
"txid" : "c3c9b69c7a5c4e6b3a5978eed0a1d24379de7206e58bce22d66d089063454d0e",
"vout" : 0,
"scriptSig" : {
"asm" :
"304402200f9e6e9bacd1f0d44525265455e92014faba5931a0ee8517664777d38c090d95022
000905089d5bfcf8509589984f9b79182ea1bcbf6e6ae16f765efd8390f3c835201
04681901c41fe94cabc8e809ca1f830fd6bc953d88254337db8ab1db9448ecd8bb2fec05f74f
38abb05f4fd5d7040f9c011365967c24672514c2a40f20dde07094",
"hex" : "4730…"
},
"sequence" : 4294967295
}
],
```

Exploring Transactions

There is one output of 5 mBTC to our new 1Dima...htKz address.


```
"vout" : [
{
"value" : 0.00500000,
"n" : 0,
"scriptPubKey" : {
"asm" : "OP_DUP OP_HASH160 8b87c4f4c177a46de7d50b7dd9840c16caa47280
OP_EQUALVERIFY OP_CHECKSIG",
"hex" : "76a9148b87c4f4c177a46de7d50b7dd9840c16caa4728088ac",
"reqSigs" : 1,
"type" : "pubkeyhash",
"addresses" : [
"1Dima5vfScYn342c7SfcX2pFYSu3rqhtKz"
]
```

Exploring Transactions

Once the transaction is confirmed, the **gettransaction** command returns additional information, showing the block hash (identifier) and block index for the block in which the transaction was included.

```
gettransaction
0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c
{
"amount" : -0.01080000,
"fee" : 0.01080000,
"confirmations" : 2,
"blockhash" :
"000000000000000002320499cc4e60f5a515a03b088925f78b728bdf79ed5ac86",
"blockindex" : 189,
"blocktime" : 1399872120,
"txid" :
"0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c",
"walletconflicts" : [
],
"time" : 1399871859,
"timereceived" : 1399871859,
"details" : [
{
…
```

Exploring Blocks

Command	Parameters	Description
getblock	<hash>	Returns information about the block with the given hash.
getblockhash	<index>	Returns the header hash of a block at the given height

Commands to use: https://chainquery.com/bitcoin-cli

Exploring Transactions

We are now going to analyze the block we obtained earlier.

```
gettransaction
0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c
{
"amount" : -0.01080000,
"fee" : 0.01080000,
"confirmations" : 2,
"blockhash" :
"000000000000000002320499cc4e60f5a515a03b088925f78b728bdf79ed5ac86",
"blockindex" : 189,
"blocktime" : 1399872120,
"txid" :
"0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c",
"walletconflicts" : [
],
"time" : 1399871859,
"timereceived" : 1399871859,
"details" : [
{
...
```

Exploring Blocks

As you can see, this block contains many transactions.

```
getblock 00000000000000002320499cc4e60f5a515a03b088925f78b728bdf79ed5ac86
{
"hash" : "00000000000000002320499cc4e60f5a515a03b088925f78b728bdf79ed5ac86",
"confirmations" : 20,
"size" : 144825,
"height" : 300323,
"version" : 2,
"merkleroot" :
"5c782440831d895dbe2851999d403f08a59768a633de027288403efa472081c8",
"tx" : [
"5a127914b627a7657759c0a09df974d11d6712bd707731e9bb6b7b675d326aeb",
"050347e8a6babdd74fc60809c29f16d1bc23f0f5dd2ac329499b57166e197e18",
"999972c1afe4c311e46e475a661dc83397cd3896c79f7cda4374d0372433de9a",
…
"161cb709e9b9e15617d0827af6282fee53484fe9ca4a575258989d8809256fd8"
],
"time" : 1399872120,
"nonce" : 1602350785,
"bits" : "1900896c",
"difficulty" : 8000872135.96816350,
"chainwork" :
"0000000000000000000000000000000000000000005cd4f1cdf66447a1f6c4",
"previousblockhash" :
"00000000000000044340d7a81d3165439ddbabc94521754f00daeaaa0aae09b",
"nextblockhash" :
"000000000000000001ad17972576160667dd6f246045ff03e50716242d68faf7"
}
```

Exploring Blocks

Our transaction can also be found in this block.

```
"998342493e6deb9efda2250878a86b42b74fbdd99365e5074d9d7517f6f92e50",
"1f74548a6b92149b2dd523928ebfc830aa80f9f01bca94f281a934e647696981",
"4fa2eb6c33189190fc515066c0a38da9573583f3b795fe2f0b75147278b3c037",
"15766c8dee6053f2d36bba568497850ca0d39d782d239816261d7bef9b3d25b3",
"0408953d670d0f268b70f449772bd3d1c1b80c690be6f3017e3f9bdaa01c0b6c",
"8062654f3f2de78537da07784a199e6724a5ea43a281b7f7536d9d2ad2a468a6",
"b90e6279b0e4bf697d71a0129bbc76e993e72c397f411b796935c56f5e58d12b",
"22246aaaa62afa3d191df40226dfcb3a64fe3e426b67525d4eb1d906a6f9ef87",
"33d20b9bbbf7ea35f070aabc8a34db51ba0fa172dcaac114fb00b28f31c8cbf3",
"0cf5e8f7fbc9dc0d853c8699abd5ce8f1ff497da8b1acc819d07a7d2ae54dc30",
```

Exploring Blocks

We can retrieve blocks by index ("block height"), where "0" is the height of the Genesis block.

The Genesis block only contains one transaction, the coinbase transaction. Prior to the first Bitcoin transaction (found on block 170), all blocks only contained the coinbase transaction.

This is how bitcoins are generated.

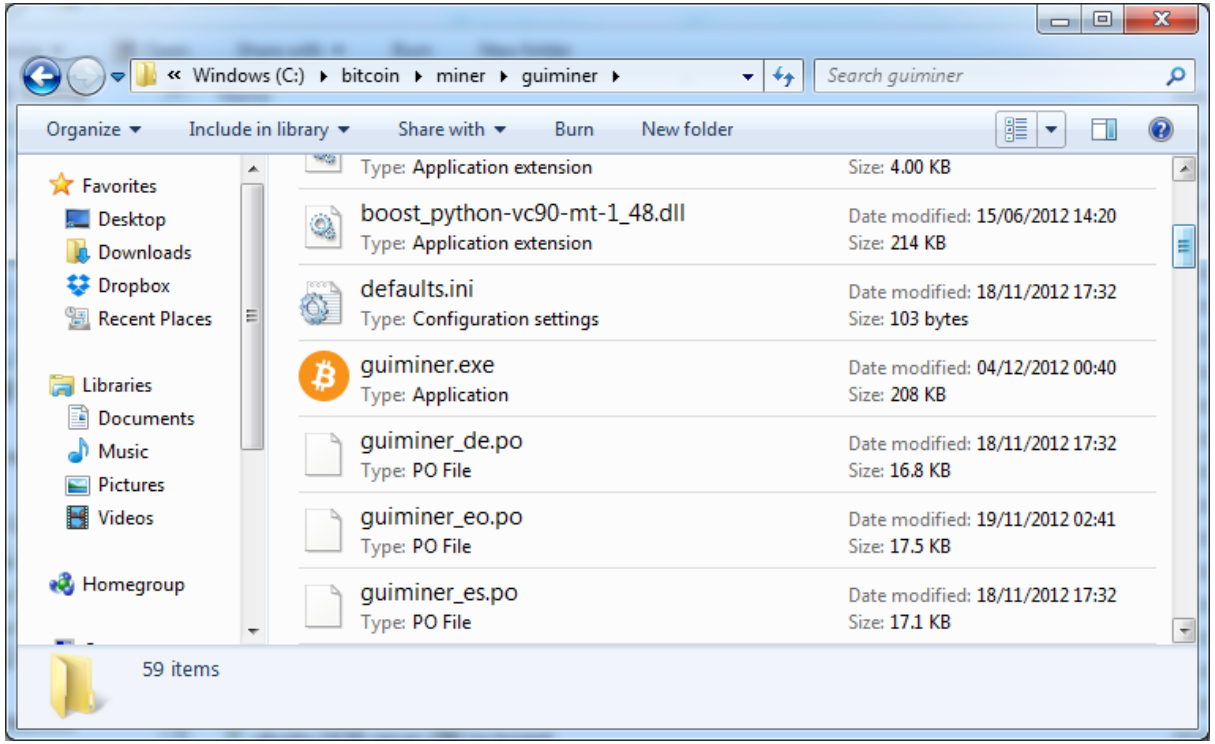
```
getblockhash 0
000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

getblock
000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
{
"hash" :
"000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f",
"confirmations" : 300344,
"size" : 285,
"height" : 0,
"version" : 1,
"merkleroot" :
"4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b",
"tx" : [
"4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b"
],
"time" : 1231006505,
"nonce" : 2083236893,
"bits" : "1d00ffff",
"difficulty" : 1.00000000,
"chainwork" :
"0000000000000000000000000000000000000000000000000100010001",
"nextblockhash" :
"00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048"
}
```

Session 5: Bitcoin in Practice 2

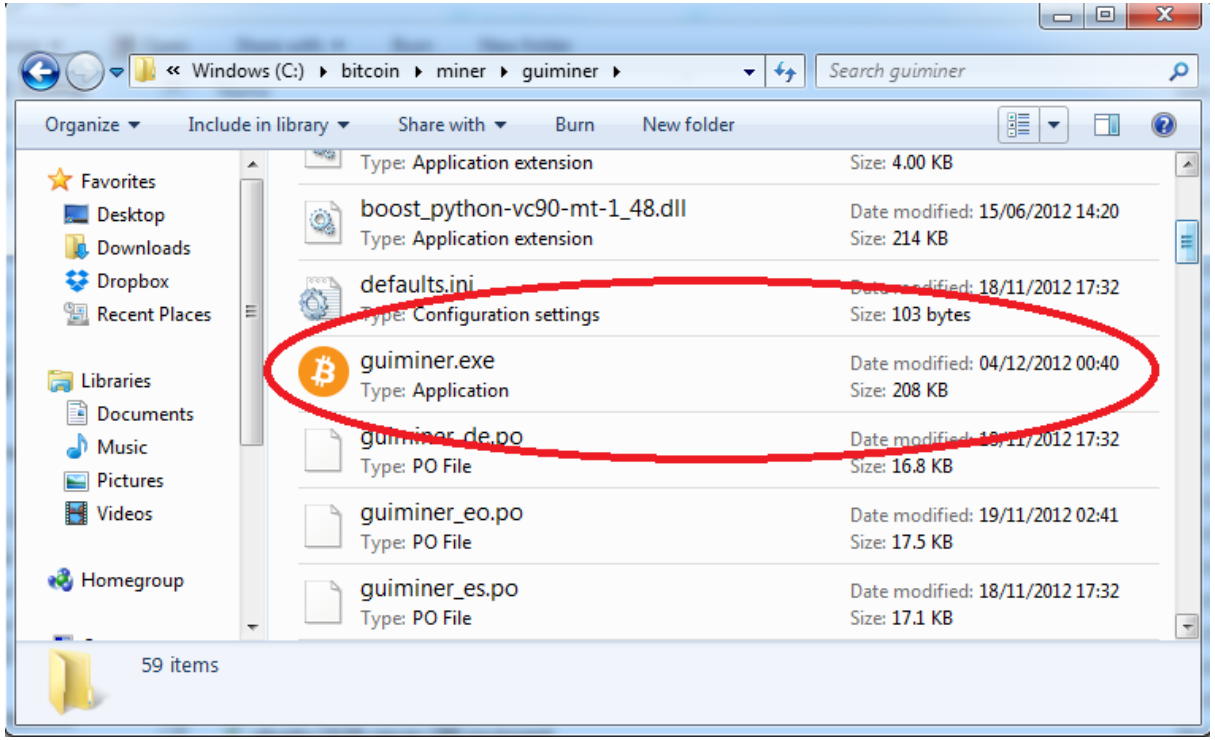
6. Mining and Mining Pools

Download a GUI-miner



- GUIMiner is a graphical frontend for pooled or solo mining, supporting ATI and NVIDIA GPUs, as well as CPUs.
- While mining this way is no longer considered practical, you can still find the software at: http://guiminer.org/.
- This will give you a self-extracting archive. When you run it, you have to specify the desired location, for example, c:\bitcoin\miner\
- After extracting, you will have a guiminer folder with the following content.

Download a GUI-miner

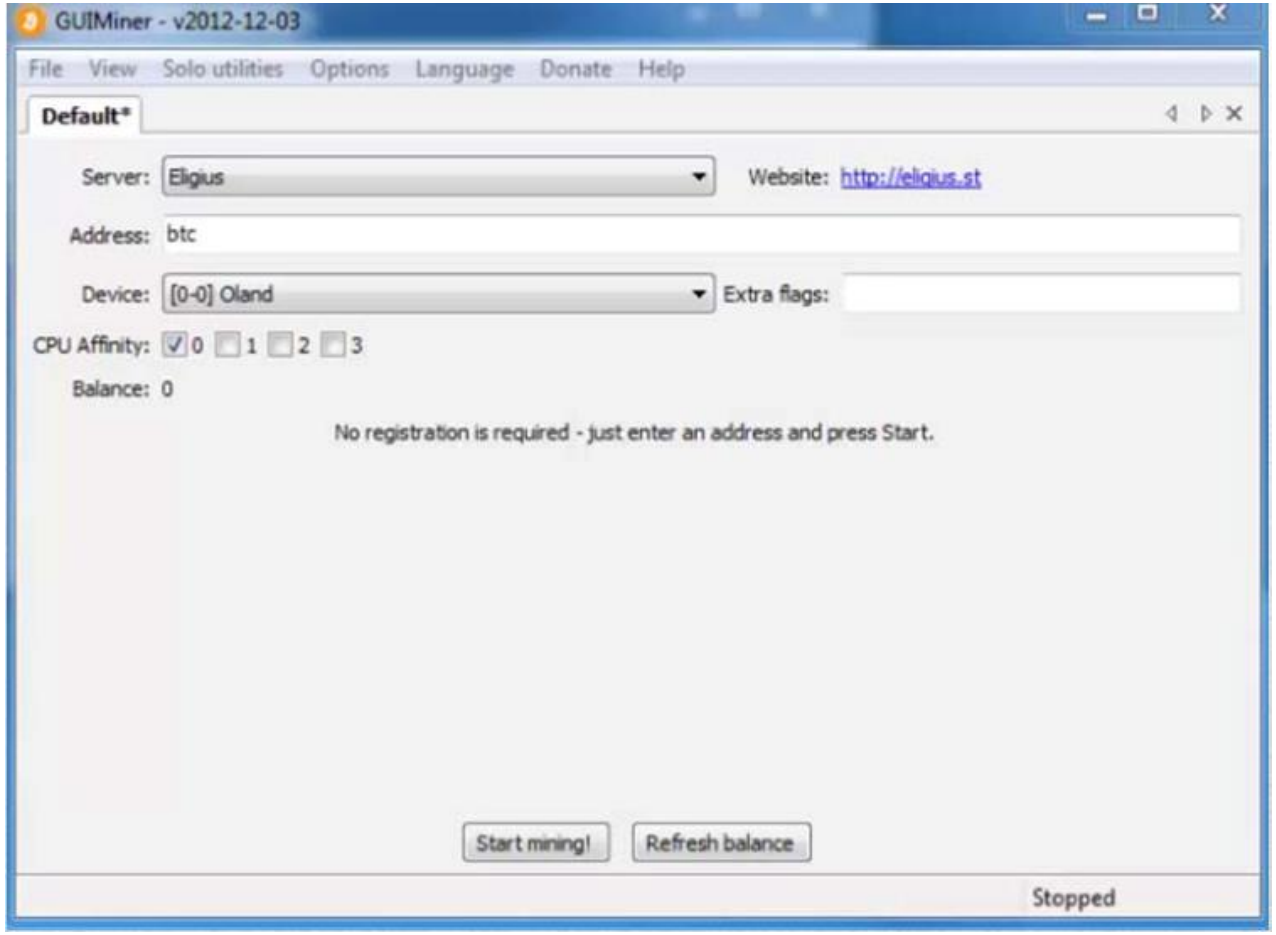


- You can create a shortcut for guiminer.exe

Solo Mining

- The difference between solo mining and pool mining was discussed earlier in Session 3.
- In order to start solo mining, you need to have Bitcoin Core installed and synchronized.
- When running in server mode, Core accepts RPC calls from other programs, including the miner. The miner can use the **getblocktemplate** RPC call.
- **getblocktemplate** returns a whole template for the next block to be generated.
- When a solution is found, the miner submits it to the network.

Solo Mining: Video illustration



- Video illustration of solo mining:
- Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=ZtHkXekq4IA

Pool mining & Video illustration

- In order to start pool mining, you need to create an account with one of the mining pool operators. We have discussed pool selection earlier in Session 3.
- In the following video illustration, we created an account with bitcoin.cz (now known as slushpool). The registration procedure depends on the chosen pool.

Video illustration of pool mining below.

Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=KRvY_Q0q7Z8

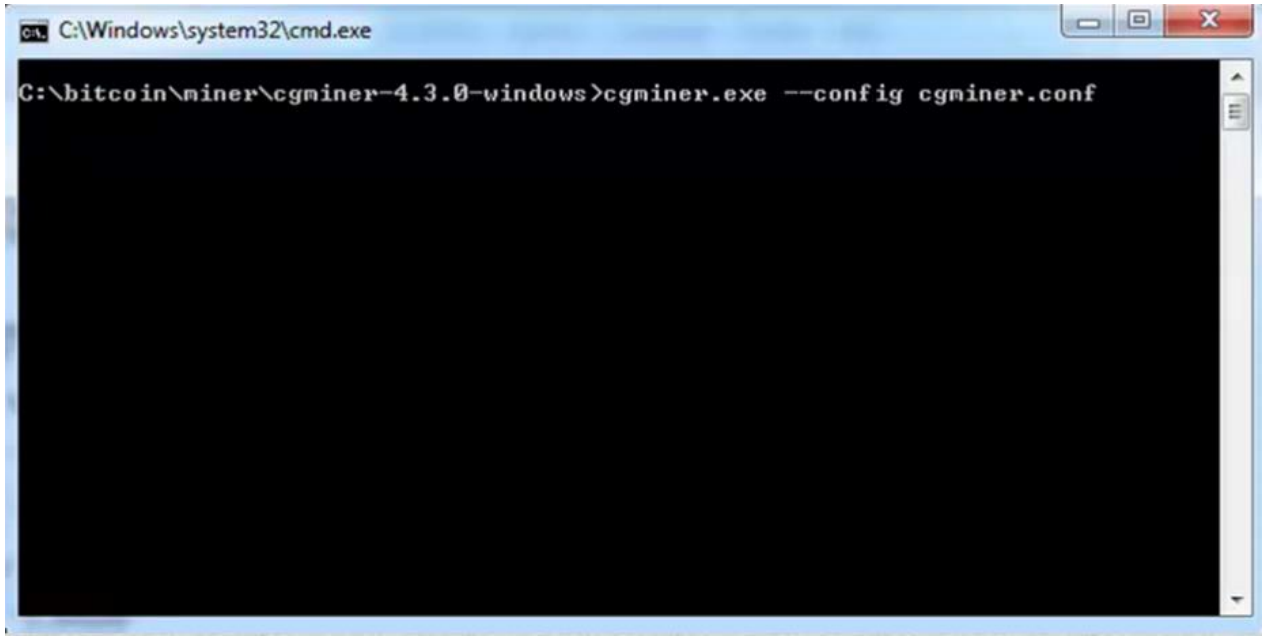
Learn More:

See No. 4 on the further reading list to see a comparison of mining pools and hardware.

Mining - ASIC Mining

- In the early days of Bitcoin, when the difficulty was much lower, CPUs were the main method to perform mining calculations.
- Today, CPU mining cannot provide any level of meaningful performance. Instead, bitcoin mining is now done using **application-specific integrated circuits** (ASICs).
- ASIC miners can be standalone devices that are connected to the network.
- These mining machines are relatively expensive and contain dozens of ASIC chips.

ASIC Mining: Video illustration



Video illustration of the miner reports, work progress and performance statistics.

Click on the image to start, or alternatively:

https://www.youtube.com/watch?v=8D9cLq9ESv8

Mining - ASIC Mining

- GPU mining is another type of mining where the GPU (Graphics Processing Unit) is used to perform mining calculations.
- GPU miners are usually based on custom-built PCs that may have up to 6 graphics cards.
- GPU mining is faster than CPU mining, and consumes a lot of power.
- Still, the significantly higher output of ASICs combined with their low energy consumption per hash (by comparison) has made them the method of choice when mining bitcoins.
- GPUs are hardly used for Bitcoin mining anymore, though they are used to mine other coins.

UNIVERSITY of NICOSIA

Questions?

Contact Us:

Twitter: **@mscdigital**
Course Support: **digitalcurrency@unic.ac.cy**
IT & Live Session Support: **dl.it@unic.ac.cy**