

rieMiner 0.9, manual

Version 2018-12-15

rieMiner is a Riecoin miner supporting both solo and pooled mining. It was originally adapted and refactored from gatra's cpuminer-rminerd (<https://github.com/gatra/cpuminer-rminerd>) and dave-andersen's fastric (<https://github.com/dave-andersen/fastric>), though there is no remaining code from rminerd anymore.

Solo mining is done using the GetBlockTemplate protocol, while pooled mining is via the Stratum protocol. A benchmark mode is also proposed to compare more easily the performance between different computers.

This present document serves as manual for rieMiner 0.9, and can be downloaded [here](#). You can also visit the [project page](#) to find the source code and build instructions.

Direct links to the latest official [Windows x64](#) and [Win32](#) standalone executables. Binaries built on Debian 9 with almost complete static linking also available (these should run on fresh Debian and Ubuntu installations): [Deb64](#) and [Deb32](#). Also note that 32 bits builds are much slower.

I hope that this program will be useful for you! The Riecoin community thanks you for your participation, you will be a contributor to the robustness of the Riecoin network. Happy mining!

I provide a Profitability Calculator [here](#).

Recommended configuration for mining

- Windows 10 or Debian 9;
- Intel Core i7 6700 or better, or AMD Ryzen R5 1600 or better;
- 8 GiB of RAM.

To run with less RAM, the prime table limit must be manually set at a lower value in the options, see below.

Run and configure this program

You can run the rieMiner executable using `./` on Linux or simply by clicking on its icon on Windows.

If no "rieMiner.conf" next to the executable was found, you will be assisted to configure rieMiner. Answer to its questions to start mining. If there is a "rieMiner.conf" file next to the executable with incorrect information that was read, you can delete this to get the assistant.

Alternatively, you can create or edit this "rieMiner.conf" file next to the executable yourself, in order to provide options to the miner. The rieMiner.conf syntax is very simple: each option is given by a line such

```
Option type = Option value
```

It is case sensitive, but spaces and invalid lines are ignored. A line starting with "`#`" will also be ignored. **Do not put ; at the end or use other delimiters than =** for each line, and **do not confuse rieMiner.conf with riecoin.conf!** If an option is missing, the default value(s) will be used. If there are duplicate lines, the last one will be used. Here is a sample configuration file for solo mining, with comments explaining the main available options.

```
# Mining mode: Solo for solo mining via GetBlockTemplate, Pool for pooled mining using
# Stratum, Benchmark for testing. Default: Benchmark
Mode = Solo

# IP and port of the Riecoin wallet/server or pool. Default: 127.0.0.1 (your computer),
# port 28332 (default port for Riecoin-Qt)
Host = 127.0.0.1
Port = 28332

# Username and password used to connect to the server (same as rpcuser and rpcpassword
# in riecoin.conf for solo mining). If using Stratum, the username includes the worker
# name (username.worker). Default: empty values
Username = user
Password = /70P$€CR€7/

# Custom payout address for solo mining (GetBlockTemplate only). Default: this donation
# address
```

```
PayoutAddress = RPtttMeDWkzjqVp62SdG2ExtCor9w54EB
```

```
# Number of threads used for mining. Default: 8  
Threads = 8
```

```
# The prime table used for mining will contain primes up to the given number.  
# Use a bigger limit if you have 16 GiB of available RAM or more, as this will reduce  
# the ratio between the n-tuple and (n + 1)-tuple counts (but also the 1-tuple find  
# rate).  
# Reduce if you have less than 8 GiB of RAM (or if you want to reduce memory usage).  
# It can go up to  $2^{64} - 1$ , but setting this at more than  $2^{33}$  will usually be too much  
# and decrease performance. Default:  $2^{31}$   
PrimeTableLimit = 2147483648
```

```
# Refresh rate of the stats in seconds. 0 to disable them and only notify when a long  
# enough tuple or share is found, or when the network finds a block. Default: 30  
RefreshInterval = 60
```

```
# For solo mining, submit not only blocks (6-tuples) but also k-tuples of at least the  
# given length. Additionally, the base prime of such tuple will be shown in the  
# Benchmark Mode. Default: 6  
TupleLengthMin = 4
```

```
# For solo mining, add consensus rules in the GetBlockTemplate RPC call, each separated  
# by a comma. Useful for softforks, for example, to mine SegWit transactions, you would  
# need the following line. Default: no rule  
# Rules = segwit
```

```
# Other options  
# BenchmarkDifficulty = 1600  
# BenchmarkTimeLimit = 0  
# Benchmark2tupleCountLimit = 100000  
# SieveBits = 25  
# SieveWorkers = 0
```

It is also possible to use custom configuration file paths, examples:

```
./rieMiner config/example.txt  
./rieMiner "config 2.conf"  
./rieMiner /home/user/rieMiner/rieMiner.conf
```

Benchmark Mode options

- `BenchmarkDifficulty` : sets the testing difficulty (must be from 265 to 32767). Default: 1600;
- `BenchmarkTimeLimit` : sets the testing duration in s. 0 for no time limit. Default: 0;
- `Benchmark2tupleCountLimit` : stops testing after finding this number of 2-tuples. 0 for no limit. Default: 50000.

Advanced options

They can be useful to get better performance depending on your computer.

- `SieveBits` : size of the segment sieve is $2^{\text{SieveBits}}$ bits, e.g. 25 means the segment sieve size is 4 MiB. Choose this so that `SieveWorkers*SieveBits` fits in your L3 cache. Default: 25;
- `SieveWorkers` : the number of threads to use for sieving. Increasing it may solve some CPU underuse problems, but will use more memory. 0 for choosing automatically based on number of `Threads` and `PrimeTableLimit`. Default: 0.

Memory problems

If you have memory errors (Unable to allocate... or Bad Allocs), try to lower the `PrimeTableLimit` value in the configuration file.

Statistics

rieMiner will regularly print some stats, and the frequency of this can be changed with the RefreshInterval parameter as said earlier.

For solo mining, rieMiner will regularly show the primes per second speed, and the 1 to 2-tuples/s ratio. From this, it will also estimate the average time to find a block (note that all the ratios are the same, and the estimation should be fairly precise). Of course, even if the average time to find a block is for example 2 days, you could find a block in the next hour as you could find nothing during a week. The number of 2 to 6-tuples found since the start of the mining is also shown.

For pooled mining, the shares per minute metric and the numbers of valid and total shares are shown instead. As it is hard to get a correct earnings estimation from k-shares, no other metric is shown. The Benchmark Mode (or solo mining) can be used to get better figures for comparisons.

rieMiner will also notify if it found a k-tuple ($k \geq$ Tuples option value) in solo mining or a share in pooled mining, and if the network found a new block. If it finds a block or a share, it will tell if the submission was accepted (solo mining only) or not. For solo mining, if the block was accepted, the reward will be generated for the address specified in the options. You can then spend it after 100 confirmations. Note that orphaned blocks will be shown as accepted.

Solo mining specific information

Note that other ways for solo mining (protocol proxies,...) were never tested with rieMiner. It was written specifically for the official wallet and the existing Riecoin pools.

Configure the Riecoin wallet for solo mining

We assume that Riecoin Core is already working and synced. To solo mine with it, you have to configure it.

- Find the riecoin.conf configuration file. It should be located in /home/username/.riecoin or equivalent in Windows;
- **Do not confuse this file with the rieMiner.conf!**
- An example of riecoin.conf content suitable for mining is

```
rpcuser=(username)
rpcpassword=(password)
rpcport=28332
port=28333
rpcallowip=127.0.0.1
connect=(nodeip)
...
connect=(nodeip)
server=1
daemon=1
```

The (nodeip) after connect are nodes' IP, you can find a list of the nodes connected the last 24 h here: <https://chainz.cryptoid.info/ric/#!network>. The wallet will connect to these IP to sync. The following always worked fine for me:

```
connect=nodes.riecoin-community.com
connect=5.9.39.9
connect=37.59.143.10
connect=78.83.27.28
connect=144.217.15.39
connect=149.14.200.26
connect=178.251.25.240
connect=193.70.33.8
connect=195.138.71.80
connect=198.251.84.221
connect=199.126.33.5
connect=217.182.76.201
```

If you wish to mine from another computer, add another rpcallowip=ip.of.the.computer, or else the connection will be refused. Choose a username and a password and replace (username) and (password).

Work control

You might have to wait some consequent time before finding a block. What if something is actually wrong and then the time the miner finally found a block, the submission fails?

First, if for some reason rieMiner disconnects from the wallet (you killed it or its computer crashed), it will detect that it has not received the mining data and then just stop mining: so if it is currently mining, everything should be fine.

If you are worried about the fact that the block will be incorrectly submitted, here comes the TupleLengthMin option. Indeed, you can send invalid blocks to the wallet (after all, it is yours), and check if the wallet actually received them and if these submissions are properly processed. When such invalid block is submitted, you can check the debug.log file in the same location as riecoin.conf, and then, you should see something like

```
ERROR: CheckProofOfWork() : n+10 not prime
```

Remember that the miner searches numbers n such that n , $n + 2$, $n + 6$, $n + 10$, $n + 12$ and $n + 16$ are prime, so if you set the TupleLengthMin option to for example 3, rieMiner will submit a n such that n , $n + 2$ and $n + 6$ are prime, but not necessarily the other numbers, so you can conclude that the wallet successfully decoded the submission here, and that everything works fine. If you see nothing or another error message, then something is wrong (possible example would be an unstable overclock)...

Also watch regularly if the wallet is correctly syncing, especially if the message “Blockheight = ...” did not appear since a very long time (except if the network is mining the superblock). In Riecoin-Qt, this can be done by hovering the check at the lower right corner, and comparing the number with the latest block found in a Riecoin explorer. If something is wrong, try to change the nodes in riecoin.conf or check your connection.

Pooled mining specific information

Existing pools:

- [XPoolX](#)
 - Host = mining.xpoolx.com
 - Port = 5000
 - Owner: [xpoolx](#) - info@xpoolx.com
 - They also support Solo mining via Stratum with a 5% fee
- [RiePool](#)
 - Host = riepool.ovh
 - Port = 8000
 - Owner: [Simba84](#) - inforiepool@gmail.com
- [uBlock.it](#)
 - Host = mine.ublock.it or mine.blockocean.com
 - Port = 5000
 - Owner: [ziiip](#) - netops.ublock.it@gmail.com
 - Invitation needed to join (contact the owner)

The miner will disconnect if it did not receive anything during 3 minutes (time out).

Benchmarking

rieMiner provides a way to test the performance of a computer, and compare with others. This feature can also be used to appreciate the improvements when trying to improve the miner algorithm. When sharing benchmark results, you must always communicate the difficulty, the prime table limit (PTL), the test duration, the CPU model, the memory speeds (frequency and CL), the miner version, and the OS. Also, do not forget to precise if you changed other options, like the SieveWorkers or Bits.

To compare two different platforms or settings, you must absolutely test with the same difficulty, during enough time. The proposed parameters, conditions and interpretations for serious benchmarking are:

- Standard Benchmark
 - Difficulty of 1600;
 - PTL of $2^{31} = 2147483648$;
 - No time limit;
 - Stop after finding 50000 2-tuples or more;
 - The computer must not do anything else during testing;

- The system must not swap. Else, the result would not make much sense. Ensure that you have enough memory when benchmarking.

The test will be fairly long, but similar to the real mining conditions. Once the benchmark finished itself (not by the user), it will print something like:

```
100000 2-tuples found, test finished. rieMiner 0.9, difficulty 1600, PTL 2147483648
BENCHMARK RESULTS: 233.354130 primes/s with ratio 28.955020 -> 0.990626 block(s)/day
```

Generally speaking, the block(s)/day metric is the one that should be shared or used to compare performance, though it is always good to also take in consideration the other ones. Moreover, for a given difficulty and PTL, the ratio should be the same, and the more precise primes/day metric can be used instead for comparisons.

The precision will be about 2 significant digits for the block(s)/day. To get 3 solid digits, about 1 million of 2-tuples would need to be found, which would be way too long to be practical for the Standard Benchmark.

A run with valid parameters for the Standard Benchmark will additionally print the message

```
VALID parameters for Standard Benchmark
```

Which should appear if you want to share your results.

You could stop before 50000 2-tuples, for example at 10000, if you just want a rough estimation of the performance. However, even after this long, the values are often still very imprecise, and can lead to confusion, like a slightly slower computer getting better results. This remark is critical for people wanting to optimize the miner.

A few results

Done with rieMiner 0.9, 100000 2-tuples. Data: primes/s, ratio -> block(s)/day.

- AMD Ryzen R7 2700X @4 GHz, DDR4 2400 CL15, Debian 9: 233.354130, 28.955020 -> 0.990626
- AMD Ryzen R7 2700X @3 GHz, DDR4 2400 CL15, Debian 9: 177.234506, 28.988780 -> 0.748018
- Intel Core i7 6700K @3 GHz, DDR4 2400 CL15, Debian 9: 89.288621, 28.883051 -> 0.383791

Miscellaneous

Unless the weather is very cold, I do not recommend to overclock a CPU for mining, unless you can do that without increasing noticeably the power consumption. My 2700X computer would draw much, much more power at 4 GHz/1.2875 V instead of 3.7 GHz/1.08125 V, which is certainly absurd for a mere 8% increase. To get maximum efficiency, you might want to find the frequency with the best performance/power consumption ratio (which could also be obtained by undervolting the processor).

If you can, try to undervolt the CPU to reduce power consumption, heat and noise.

Developers and license

- [Pttn](#), author and maintainer, contact: dev at Pttn dot me

Parts coming from other projects and libraries are subject to their respective licenses. Else, this work is released under the MIT license.

Notable contributors

- [Michael Bell](#): assembly optimizations, improvements of work management between threads, and some more.

Versioning

The version naming scheme is 0.9, 0.99, 0.999 and so on for major versions, analogous to 1.0, 2.0, 3.0, . . . The first non 9 decimal digit is minor, etc. For example, the version 0.9925a can be though as 2.2.5a. A perfect bug-free software will be version 1.