Rigel Miner Command Reference

-a, --algorithm <ALGORITHM>

Selects the mining algorithm

Currently supported:

abelian	(ABEL)
alephium	(ALPH)
autolykos2	(ERGO)
etchash	(ETC)
ethash	(ETHW, XPB, OCTA, etc.)
ethashb3	(HYP)
fishhash	(IRON)
ironfish	(pre-fork IRON)
karlsenhash	(KLS)
kawpow	(RVN, AIPG, XNA, CLORE, NEOX, etc.)
nexapow	(NEXA)
octopus	(CFX)
pyrinhash	(PYI)
sha256ton	(GRAM)
sha512256d	(RXD)
xelishash	(pre-fork XEL)
xelishashv2	(XEL)
zil	(ZIL)

To dual or triple mine pass the algorithm list

separated by `+`, e.g. `octa+karlsenhash+zil`

--a1 <ALGORITHM>

Sets the primary algorithm (alternative to `-a`)

--a2 <ALGORITHM>

Sets the second algorithm (alternative to `-a`)

--a3 <ALGORITHM>

Sets the third algorithm (alternative to `-a`)

--coin <COIN>

Sets the coin ticker

Only applicable to DAG based algorithms used by multiple coins.

Unless you're mining to a profit switching pool like Nicehash,

setting the coin ticker has the benefit of eliminating DAG rebuilds when

the miner switches to dev fee and back to the user job.

Note that not all possible coins are recognised by the miner,

and the up-to-date list is maintained on the project's webpage.

Supported coins (the list is not exhaustive):

`kawpow`: aipg, clore, neox, xna, rvn

`ethash`: ethw, octa, xpb

```
    When dual or triple mining the value should be prepended with
    the algorithm index `[<index>]`. Primary algorithm has index 1.

    Examples:
    --coin rvn
    --coin aipg
    --coin [1]octa

-o, --url <URL>
    Sets the pool URL

    Format:
    <mining protocol>+<transport protocol>://<pool hostname>:<port number>

    Mining protocols:
      stratum
      ethproxy
      ethstratum
      zmp (zil only)
      solo (xelishash only)
    Transport protocols: tcp, ssl

    When dual or triple mining the value should be prepended with
    the algorithm index `[<index>]`. Primary algorithm has index 1.
```

Failover pools are set by using `-o` multiple times.

Examples:

-o stratum+ssl://pool.woolypooly.com:3112

-o zmp+tcp://us1-zil.shardpool.io:3333

mine Zilliqa to shardpool using ZMP protocol

-o [1]stratum+tcp://eth.f2pool.com:6688 -o [2]stratum+ssl://pool.woolypooly.com:3112

mine the primary algorithm to f2pool and the second algorithm to woolypooly

-u, --username <USERNAME>

Sets the username for pool authorisation

When dual or triple mining the value should be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

-p, --password <PASSWORD>

Sets the password for pool authorisation

When dual or triple mining the value should be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

-w, --worker <WORKER>

Sets the worker name

When dual or triple mining the value should be prepended with

```
        the algorithm index `[<index>]`. Primary algorithm has index 1.

  -d, --devices <D1,D2,...>
        Sets GPU devices to use for mining

        Must be a comma-separated list of device indices. If not specified,
        all detected devices will be used. Devices are ordered by their
        PCI bus addresses. First GPU has index of 0.
        To display all available devices, use --list-devices.

        Example:
        -d 0,2,4,5

      --list-devices
        List available mining devices and exit

      --kernel <KERNEL1,KERNEL2,...>
        Sets CUDA kernel for selected algorithm(s)

        Some algorithms have multiple implementations with different characteristics,
        for example, one may be more energy efficient and another can achieve
        higher hashrate at a cost of increased power consumption. In such cases
        the user can choose the implementation that suits their requirements.

        Supported algorithms and values:
```

```
    nexapow
      1 - default kernel
      2 - some 20xx and 30xx GPUs may benefit from choosing this kernel
        as more energy efficient than #1 when targeting low power draw

    sha256ton
      1 - default kernel
      2 - some 40xx GPUs may benefit from choosing this kernel
        as more energy efficient than #1 when targeting low power draw

    Comma-separated list of values can be used to set values per-GPU
    To skip a GPU, set the corresponding value to underscore `_`.
    When dual or triple mining the value may be prepended with
    the algorithm index `[<index>]`. Primary algorithm has index 1.

    Example:
    --kernel _,_,2,2
      GPU#0 and GPU#1 will use their respective default kernels
      GPU#1 and GPU#2 will use kernel #2

  --tune <TUNE1,TUNE2,...>
    Sets tuning values for selected algorithm(s)

    Supported algorithms:
      xelishash
```

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`.
When dual or triple mining the value may be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

Example:

```
--tune 184,_,320
    GPU#0 will use tuning value 184
    GPU#1 will be auto-tuned
    GPU#2 will use tuning value 320
```

--dual-mode <MODE1,MODE2,...>

Controls GPU behaviour in dual mining mode

Format: <algo>:<dual ratio>
`<algo>` must be one of the following:

```
    a1  - first algorithm
    a2  - second algorithm,
    a12 - both
```

`<dual ratio>` is optional, and, if present,
must be
`rXX` (`dual ratio` coefficient set to `XX`), or
`hXX` (first algorithm hashrate will drop to `XX`%)

```
    Comma-separated list of values can be used to set values per-GPU
    To skip a GPU, set the corresponding value to underscore `_`

    Example:
    --dual-mode a1,_,a12:r5.2,a12:h92,a2
      GPU#0 will mine the primary algorithm
      GPU#1 will dual mine with default settings (primary algorithm at 95%)
      GPU#2 will dual mine, dual ratio set to 5.2
      GPU#3 will dual mine, primary algorithm hashrate at 92%
      GPU#4 will mine the second algorithm

  --temp-limit <LIMIT1,LIMIT2,...>
    Sets GPU thermal limits

    Sets the temperatures of GPU core and memory upon reaching which the miner
    stops the overheated GPU and waits for it to cool down before resuming.

    Comma-separated list of values can be used to set values per-GPU
    To skip a GPU, set the corresponding value to underscore `_`

    Examples:
    --temp-limit tc[60-70]
      stops the GPU when its core temperature reaches 70 degrees celsius
      and resumes mining when it cools down to 60 degrees
```

--temp-limit tm[105-115]

stops the GPU when its memory temperature reaches 115 degrees celsius

and resumes mining when it cools down to 60 degrees

--temp-limit tc[60-70]tm[105-115]

enables both core and memory temperature limits

--cpu-check

Enables CPU verification of found shares before sending them to the pool

--hashrate-avg <SECONDS>

Hashrate averaging window in seconds. Default is 10.

--power-avg <SECONDS>

Power usage averaging window in seconds. Default is 10.

--cclock <FREQ1,FREQ2,...>

Sets GPU core clock frequency offset in MHz

Comma-separated list of values can be used to set values per-GPU

To skip a GPU, set the corresponding value to underscore `_`.

When dual or triple mining the value may be prepended with

the algorithm index `[<index>]`. Primary algorithm has index 1.

--mclock <FREQ1,FREQ2,...>

Sets GPU memory clock frequency offset in MHz

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`.
When dual or triple mining the value may be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

--lock-cclock <FREQ1,FREQ2,...>

Locks GPU core clock frequency to the given value in MHz

Comma-separated list of values can be used to set values per-GPU
Set to `X` to reset the locked clock (unlock)
To skip a GPU, set the corresponding value to underscore `_`.
When dual or triple mining the value may be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

--lock-mclock <FREQ1,FREQ2,...>

Locks GPU memory clock frequency to the given value in MHz

Comma-separated list of values can be used to set values per-GPU
Set to `X` to reset the locked clock (unlock)
To skip a GPU, set the corresponding value to underscore `_`.
When dual or triple mining the value may be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

--pl <PL1,PL2,...>

Sets GPU power limit to the given value in Watts

Comma-separated list of values can be used to set values per-GPU

To skip a GPU, set the corresponding value to underscore `_`.

When dual or triple mining the value may be prepended with

the algorithm index `[<index>]`. Primary algorithm has index 1.

--mt <MT1,MT2,...>

Sets memory tweaks for Pascal GPUs with GDDR5/GDDR5X memory

Possible values: 1, 2, 3, 4, 5, 6 (REFRESH will be set to `mt * 16`)

Comma-separated list of values can be used to set values per-GPU

To skip a GPU, set the corresponding value to underscore `_`.

When dual or triple mining the value may be prepended with

the algorithm index `[<index>]`. Primary algorithm has index 1.

--fan-control <FAN1,FAN2,...>

Sets GPU fan control mode

Can either force the GPU fans to work at a given static speed in %,

or adjust the fan speed to keep the GPU temperature (core, memory, or both)

at a desired level (auto-fan mode)

```
Format:
  --fan-control N
    set fan speed to `N`%
  --fan-control t:[Tc;Tm][Finit;Fmin-Fmax]
    enables auto-fan with `Tc` and `Tm` core and memory temperatures
    respectively, set initial fan speed to `Finit`% and limit the
    fan speed range to [`Fmin`%,`Fmax`%] range.
    every value is optional and can be set to underscore `_`
    meaning the default value should be used.
  --fan-control X
    let the driver/OS manage the fan speed (auto)

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`.
When dual or triple mining the value may be prepended with
the algorithm index `[<index>]`. Primary algorithm has index 1.

Examples:
--fan-control 30
  sets GPU fan speed to 30%
--fan-control t:[60;_][50;30-95]
  target core temp is 60 degrees, initial fan speed is 50%, 30%-95% range
--fan-control t:[_;90][_;_-95]
  target memory temp is 90 degrees, max fan speed 95%
```

--fan-control t:[70;100][_;_-_]

adjust fan so that core temperature <= 70, memory temperature <= 100,
with no restrictions on the fan speed

--dag-reset-mclock <on/off>

Enables or disables resetting memory overclock during DAG generation. Default is "on".

If enabled, the miner sets the GPU memory clock offset to 0 MHz before building the DAG to avoid its corruption, and re-applies the original offset after the DAG has been built.

Example:

--dag-reset-mclock off

disables memory OC reset (useful when OC is controlled by an external tool)

--reset-oc

Reset all overclock (including fan control) to default settings and exit

--autolykos2-prebuild <on/off>

Enables or disables autolykos2 dataset prebuild. Default is "on".

If the prebuild is enabled for a GPU but there is not enough free memory to hold two datasets, the miner will fall back to using a single dataset and prebuild will be disabled.

Comma-separated list of values can be used to set values per-GPU

To skip a GPU, set the corresponding value to underscore `_`

Example:

--autolykos2-prebuild on,off,on

enables prebuild for GPU#0 and GPU#2 and disables for GPU#1

--ethash-cache-dag <on/off>

Enables or disables ethash DAG caching. Default is "on".

If enabled, the miner attempts to keep DAGs for different epochs in memory to avoid rebuilding them in the future if another job with the same epoch is received again.

The only limiting factor of how many DAGs can be kept is the amount of memory on the GPU.

Comma-separated list of values can be used to set values per-GPU

To skip a GPU, set the corresponding value to underscore `_`

Example:

--ethash-cache-dag on,off,on

enables DAG cache for GPU#0 and GPU#2 and disables for GPU#1

--kawpow-cache-dag <on/off>

Enables or disables kawpow DAG caching. Default is "on".

If enabled, the miner attempts to keep DAGs for different epochs in memory to avoid rebuilding them in the future if another job with the same epoch is received again.
The only limiting factor of how many DAGs can be kept is the amount of memory on the GPU.

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`

Example:
--kawpow-cache-dag on,off,on
enables DAG cache for GPU#0 and GPU#2 and disables for GPU#1

--nexapow-small-lut <on/off>
Enforces using small LUT for Nexa. Default is "off".

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`

Example:
--nexapow-small-lut on,off,on
enforces small LUT for GPU#0 and GPU#2

--zil <on/off>

Enables or disables mining Zilliqa per GPU. Default is "on".

Useful for disabling ZIL mining for individual GPUs

Example:
--zil off,on,on
disables ZIL mining for GPU#0

--zil-cache-dag <on/off>

Enables or disables caching Zilliqa DAG. Default is "on".

If enabled, ZIL DAG is generated when the miner starts up and
is kept in GPU memory at all times even if it means lower performance
for the primary algorithm in dual/triple mining mode.
If disabled, the DAG is built on-demand when a ZIL round starts
and then destroyed after it's finished.

Comma-separated list of values can be used to set values per-GPU
To skip a GPU, set the corresponding value to underscore `_`

Example:
--zil-cache-dag on,off,on
enables DAG cache for GPU#0 and GPU#2 and disables for GPU#1

--zil-countdown

Enables Zilliqa countdown timer

--zil-retain-hashrate <on/off>

Enables or disables displaying ZIL hashrate between rounds. Default is "on".

--zil-test

Enables Zilliqa mining test mode

In this mode the miner will connect to a virtual Zilliqa pool which initiates short ZIL mining sessions more frequently than a normal pool (every 30s). Useful for making sure the overclock settings are stable when switching from the primary algorithm to `zil` and back - no need to wait for the next ZIL window which may start in an hour.

--zil-test-intervals <ZIL_TEST_INTERVALS>

Sets durations of primary algorithm and Zilliqa sessions in test mode

Default value: `[30;30]`

Format:

```
--zil-test-intervals [D1;D2]
    set primary algo duration to `D1` seconds
    set ZIL duration to `D2` seconds
```

Example:

```
        --zil-test-intervals [10;60]
            the miner will mine the primary algorithm for 10 seconds, switch to ZIL,
            mine ZIL for 60 seconds, switch back, and so on

    --enable-fork
        Enables algorithm switch for hard forks

    --activate-fork
        Activates algorithm switch for hard forks

        Can be used to simulate post-hard fork behaviour and test
        overclock settings, stratum connectivity etc. before the fork

-l, --log-file <LOG_FILE>
        Enables logging output of the miner to the specified log file

        If the log file already exists it will be appended.
        The miner uses log rotation keeping maximum of 5 log files,
        size limit is 5MB.

        Supports the following template parameters:
            `{algo}` - algorithm name
            `{ts}` - timestamp in the format "yyyyMMdd_HHmmss"
                    note when using this option the miner will be creating
                    a new log file every time it is launched
```

Example:

```
--log-file rigel_{algo}_{ts}.log
```

Files created:

```
    rigel_kawpow_20240115_123240.log
    rigel_kawpow_20240116_185402.log
    rigel_kawpow_20240116_190938.log
```

--log-network

Enables logging network traffic (useful for debugging)

--api-bind <IP:PORT>

Enables HTTP API and binds it to the specified socket address

Examples:

```
--api-bind 127.0.0.1:5000
--api-bind 0.0.0.0:5000
```

--proxy <PROXY>

Sets SOCKS5 proxy for all network connections including DNS lookups

Format:

No auth: <host>:<port>

Auth: <username>:<password>@<host>:<port>

```
    Examples:
    --proxy 127.0.0.1:8091
    --proxy admin:qwerty@myproxy.server.com:8080

--dns-over-https
    Enables pool DNS resolution using DNS-over-HTTPS (Cloudflare)

--no-strict-ssl
    Disables SSL/TLS certificate verification

    Useful with self-hosted mining solutions where the mining pool
    provides a self-signed certificate.
    However, adding the certificate to the system's trust store
    should be preferred (see https://github.com/rigelminer/rigel/issues/130).

--stats-interval <SECONDS>
    GPU stats reporting interval in seconds. Default is 20.
    Set to `X` to disable periodic reports.

--long-timestamps
    Enables milliseconds timestamps in the miner output

--no-pool-url
    Hides pool URL from the share submissions log
    and displays the algorithm name instead

--no-colour
    Disables colour in the miner output (TUI would still display colours)

--no-tui
    Disables terminal user interface (TUI)

--multi-device-arg-mode <MULTI_DEVICE_ARG_MODE>
    Defines argument validation in multi-GPU mode. Default is "match".

    Applicable to all per-GPU arguments (`--cclock`, `--mclock`, `--temp-limit`,
    `--kernel` etc.) when the number of supplied values is more than one and
    doesn't match the number of GPUs - this usually occurs when one of the GPUs
    disappears from the system (taken out for maintenance, PCIE connectivity loss
    and so on) but the miner is run with the same config.

    Accepted values:
        match - display an error and exit if the number of values doesn't match
                the number of GPUs exactly
        first - ignore any extra values if too many, and use the first value in the list
                for the GPUs that don't have a value
        last  - ignore any extra values if too many, and use the last value in the list
                for the GPUs that don't have a value
    Example (assuming 3 GPUs available):
      --multi-device-arg-mode first
        "--lock-cclock 1100,1200,1300,1400" will result in
          GPU#0 1100
          GPU#1 1200
          GPU#2 1300
        "--lock-cclock 1100,1200" will result in
          GPU#0 1100
          GPU#1 1200
          GPU#2 1100

  --no-watchdog
    Disables miner watchdog

-h, --help
    Print help information (use `-h` for a summary)

-V, --version
    Print version information
```