



# Multi Router Traffic Grapher -- Configuration File Format

This file is for use with mrtg-2.5.3

Note:

- Keywords must start at the beginning of a line.
  - Lines which follow a keyword line which do start with a blank are appended to the keyword line
  - Empty Lines are ignored
  - Lines starting with a # sign are comments.
- 

## Global Configuration

Workdir specifies where the logfiles and the webpages should be created.

Example:

```
WorkDir: /usr/tardis/pub/www/stats/mrtg
```

---

## Optional Global Parameters

---

### Refresh

How many seconds apart should the browser (Netscape) be instructed to reload the page? If this is not defined, the default is 300 seconds (5 minutes).

Example:

```
Refresh: 600
```

---

## Interval

How often do you call mrtg? The default is 5 minutes. If you call it less often, you should specify it here. This does two things:

- the generated HTML page does contain the right information about the calling interval ...
- a META header in the generated HTML page will instruct caches about the time to live of this page .....

In this example we tell mrtg that we will be calling it every 10 minutes. If you are calling mrtg every 5 minutes, you can leave this line commented out.

Example:

```
Interval: 10
```

---

## WriteExpires

With this switch mrtg will generate .meta files for CERN and Apache servers which contain Expiration tags for the html and gif files. The \*.meta files will be created in the same directory as the other files, so you might have to set ``MetaDir ." in your srm.conf file for this to work

NOTE: If you are running Apache-1.2 you can use the mod\_expire to achieve the same effect ... see the file htaccess-dist

Example:

```
WriteExpires: Yes
```

---

## IconDir

If you want to keep the mrtg icons in some place other than the working directory, use the *IconDir* variable to give its url.

Example:

```
IconDir: /mrtgicons/
```

---

# Configuration for each Target you want to monitor

The configuration keywords *Target* must be followed by a unique name. This will also be the name used for the webpages, logfiles and gifs created for that target.

Note that the *Target* sections can be auto-generated with the **cfgmaker** tool. Check *readme.html* for instructions.

---

## Target

With the *Target* keyword you tell mrtg what it should monitor. The *Target* keyword takes arguments in a wide range of formats:

- The most basic format is ``port:community@router" This will generate a traffic graph for the interface 'port' of the host 'router' (dns name or IP address) and it will use the community 'community' (snmp password) for the snmp query.

Example:

```
Target[ezwf]: 2:public@wellfleet-fddi.ethz.ch
```

- Sometimes you are sitting on the wrong side of the link, and you would like to have mrtg report Incoming traffic as outgoing and vice versa. This can be achieved by adding the '-' sign in front of the ``Target" description. It flips the incoming and outgoing traffic rates.

Example:

```
Target[ezci]: -1:public@ezci-ether.ethz.ch
```

- You can also explicitly define the OID to query by using the following syntax 'OID\_1&OID\_2:community@router' The following example will retrieve error counts for input and output on interface 1. MRTG needs to graph two variables, so you need to specify two OID's such as temperature and humidity or error input and error output.

Example:

```
Target[ezwf]: 1.3.6.1.2.1.2.2.1.14.1&1.3.6.1.2.1.2.2.1.20.1:public@myrouter
```

- MRTG knows a number of symbolical SNMP variable names. See the file mibhelp.txt for a list of known names. One example are the ifInErrors and ifOutErrors. This means you can specify the above as:

Example:

Target[ezwf]: ifInErrors.1&ifOutErrors.1:public@myrouter

- In all places where ``community@router" is accepted, you can add additional parameters for the SNMP communication using colon-separated suffixes. The full syntax is as follows:
- community@router[:port[:timeout[:retries[:backoff]]]]

where the meaning of each parameter is as follows:

**port**

the UDP port under which to contact the SNMP agent (default: 161)

**timeout**

initial timeout for SNMP queries, in seconds (default: 2.0)

**retries**

number of times a timed-out request will be retried (default: 5)

**backoff**

factor by which the timeout is multiplied on every retry (default: 1.0).

A value that equals the default value can be omitted. Trailing colons can be omitted, too.

Example:

Target[ezci]: 1:public@ezci-ether.ethz.ch:9161::4

This would refer to the input/output octet counters for the interface with *ifIndex 1* on *ezci-ether.ethz.ch*, as known by the SNMP agent listening on UDP port 9161. The standard initial timeout (2.0 seconds) is used, but the number of retries is set to four. The backoff value is the default.

- if you want to monitor something which does not provide data via snmp you can use some external program to do the data gathering.

The external command must return 4 lines of output:

**Line 1**

current state of the first variable, normally 'incoming bytes count'

**Line 2**

current state of the second variable, normally 'outgoing bytes count'

**Line 3**

string (in any human readable format), telling the uptime of the target.

**Line 4**

string, telling the name of the target.

Depending on the type of data your script returns you might want to use the 'gauge' or 'absolute' arguments for the *Options* keyword.

Example:

```
Target[ezwf]: `/usr/local/bin/df2mrtg /dev/dsk/c0t2d0s0`
```

Note the use of the backticks (`), not apostrophes (') around the command.

- You can also use several statements in a mathematical expression. This could be used to aggregate both B channels in an ISDN connection or multiple T1s that are aggregated into a single channel for greater bandwidth. Note the whitespace around the target definitions.

Example:

```
Target[ezwf]: 2:public@wellfleetA + 1:public@wellfleetA  
              * 4:public@ciscoF
```

---

## RouterUptime

In cases where you calculate the used bandwidth from several interfaces you normally don't get the router uptime and router name displayed on the web page.

If these interfaces are on the same router and the uptime and name should be displayed nevertheless you have to specify its community and address again with the *RouterUptime* keyword.

Example:

```
Target[kacisco.comp.edu]: 1:public@194.64.66.250 + 2:public@194.64.66.250  
RouterUptime[kacisco.comp.edu]: public@194.64.66.250
```

---

## MaxBytes

The maximum value either of the two variables monitored are allowed to reach. For monitoring router traffic this is normally specified in bytes per second this interface port can carry.

If a number higher than *MaxBytes* is returned, it is ignored. Also read the section on *AbsMax* for further info. The *MaxBytes* value is also used in calculating the Y range for unscaled graphs (see the section on *Unscaled*).

Since most links are rated in bits per second, you need to divide their maximum bandwidth (in bits) by eight (8) in order to get bytes per second. This is very important to make your unscaled graphs display realistic information. T1 = 193000, 56K = 7000, Ethernet = 1250000. The *MaxBytes* value will be used by mrtg to decide whether it got a valid response from the router.

Example:

```
MaxBytes[ezwf]: 1250000
```

---

## Title

Title for the HTML page which gets generated for the graph.

Example:

Title[ezwf]: Traffic Analysis for Our Nice Company

---

## PageTop

Things to add to the top of the generated HTML page. Note that you can have several lines of text as long as the first column is empty.

Note that the continuation lines will all end up on the same line in the html page. If you want linebreaks in the generated html use the '\n' sequence.

Example:

PageTop[ezwf]: <H1>Traffic Analysis for ETZ C95.1</H1>  
Our Campus Backbone runs over an FDDI line\nwith a maximum transfer rate of 12.5 megabytes per  
Second.

---

## Optional Target Configuration Tags

---

### AddHead

Use this tag like the *PageTop* header, but its contents will be added between </TITLE> and </HEAD>.

Example:

AddHead[ezwf]: <link rev="made" href="<mailto:mrtg@blabla.edu>";>

---

### AbsMax

If you are monitoring a link which can handle more traffic than the *MaxBytes* value. Eg, a line which uses compression or some frame relay link, you can use the *AbsMax* keyword to give the absolute maximum value ever to be reached. We need to know

this in order to sort out unrealistic values returned by the routers. If you do not set *AbsMax*, mrtg will ignore values higher than *MaxBytes*.

Example:

```
AbsMax[ezwf]: 2500000
```

---

## Unscaled

By default each graph is scaled vertically to make the actual data visible even when it is much lower than *MaxBytes*. With the *Unscaled* variable you can suppress this. It's argument is a string, containing one letter for each graph you don't want to be scaled: d=day w=week m=month y=year. In the example scaling for the yearly and the monthly graph are suppressed.

Example:

```
Unscaled[ezwf]: ym
```

---

## WithPeak

By default the graphs only contain the average values of the monitored variables - normally the transfer rates for incoming and outgoing traffic. The following option instructs mrtg to display the peak 5 minute values in the [w]eekly, [m]onthly and [y]early graph. In the example we define the monthly and the yearly graph to contain peak as well as average values.

Examples:

```
WithPeak[ezwf]: ym
```

---

## Suppress

By default mrtg produces 4 graphs. With this option you can suppress the generation of selected graphs. The option value syntax is analogous to the above two options. In this example we suppress the yearly graph as it is quite empty in the beginning.

Example:

```
Suppress[ezwf]: y
```

---

## Directory

By default, mrtg puts all the files that it generates for each target (the GIFs, the HTML page, the log file, etc.) in *WorkDir*.

If the *Directory* option is specified, the files are instead put into a directory under *WorkDir*. (For example the *Directory* option below would cause all the files for a target ezwf to be put into directory /usr/tardis/pub/www/stats/mrtg/ezwf/ .)

The directory must already exist; mrtg will not create it.

Example:

```
WorkDir: /usr/tardis/pub/www/stats/mrtg
Directory[ezwf]: ezwf
```

---

## XSize and YSize

By default mrtgs graphs are 100 by 400 pixels wide (plus some more for the labels. In the example we get almost square graphs ...

Note: XSize must be between 20 and 600; YSize must be larger than 20

Example:

```
XSize[ezwf]: 300
YSize[ezwf]: 300
```

---

## XZoom and YZoom

If you want your graphs to have larger pixels, you can ``Zoom" them.

Example:

```
XZoom[ezwf]: 2.0
YZoom[ezwf]: 2.0
```

---

## XScale and YScale

If you want your graphs to be actually scaled use *XScale* and *YScale*. (Beware while this works, the results look ugly (to be frank) so if someone wants to fix this: patches are welcome.

Example:

```
XScale[ezwf]: 1.5
YScale[ezwf]: 1.5
```



---

## Step

Change the default step from 5 \* 60 seconds to something else (I have not tested this well ...)

Example:

```
Step[ezwf]: 60
```

---

## Options

The *Options* Keyword allows you to set some boolean switches:

### **growright**

The graph grows to the left by default. This option flips the direction of growth causing the current time to be at the right edge of the graph and the history values to the left of it.

### **bits**

All the monitored variable values are multiplied by 8 (i.e. shown in bits instead of bytes) ... looks much more impressive :-) It also affects the 'factory default' labeling and units for the given target.

### **noinfo**

Suppress the information about uptime and device name in the generated webpage.

### **nopercent**

Don't print usage percentages

### **gauge**

Treat the values gathered from target as absolute and not as ever incrementing counters. This would be useful to monitor things like disk space, processor load, temperature, and the like ...

In the absence of 'gauge' and 'absolute' options, MRTG treats variable as a counter and calculates the difference between the current and the previous value and divides that by the elapsed time between the last two readings to get the value to be plotted.

### **absolute**

This is for data sources which reset their value when they are read. This means that rateup has not to build the difference between this and the last value read from the data source. The value obtained is still divided by the elapsed time between the last two readings, which makes it different from the 'gauge' option. Useful for external data gatherers.

Example:

```
Options[ezwf]: growright, bits
```

---

## Colours

The *Colours* tag allows you to override the default colour scheme. Note: All 4 of the required colours must be specified here. The colour name ('Colourx' below) is the legend name displayed, while the RGB value is the real colour used for the display, both on the graph and in the html doc.

Format is:

Colour1#RRGGBB,Colour2#RRGGBB,Colour3#RRGGBB,Colour4#RRGGBB

### **Colour1**

First variable (normally Input) on default graph

### **Colour2**

Second variable (normally Output) on default graph

### **Colour3**

Max first variable (input)

### **Colour4**

Max second variable (output)

### **RRGGBB**

2 digit hex values for Red, Green and Blue

Example:

Colours[ezwf]: GREEN#00eb0c,BLUE#1000ff,DARK GREEN#006600,VIOLET#ff00ff

---

## Background

With the *Background* tag you can configure the background colour of the generated HTML page

Example:

Background[ezwf]: #a0a0a0a

---

## YLegend, ShortLegend, Legend[1234]

The following keywords allow you to override the text displayed for the various legends of the graph and in the HTML document

### **YLegend**

The Y-axis label of the graph. Note that a text which is too long to fit in the graph will be silently ignored.

### **ShortLegend**

The units string (default 'b/s') used for Max, Average and Current

### Legend[1234IO]

## The strings for the colour legend

Example:

YLegend[ezwf]: Bits per Second  
ShortLegend[ezwf]: b/s  
Legend1[ezwf]: Incoming Traffic in Bits per Second  
Legend2[ezwf]: Outgoing Traffic in Bits per Second  
Legend3[ezwf]: Maximal 5 Minute Incoming Traffic  
Legend4[ezwf]: Maximal 5 Minute Outgoing Traffic  
LegendI[ezwf]: &nbsp;   In:  
LegendO[ezwf]: &nbsp;   Out:

Note, if *LegendI* or *LegendO* are set to an empty string with

LegendO[ezwf]:

The corresponding line below the graph will not be printed at all.

# Timezone

If you live in an international world, you might want to generate the graphs in different timezones. This is set in the TZ variable. Under certain operating systems like Solaris, this will provoke the localtime call to give the time in the selected timezone ...

Example:

Timezone[ezwf]: Japan

The Timezone is the standard Solaris timezone, ie Japan, Hongkong, GMT, GMT+1 etc etc.

## Weekformat

By default, `mrtg` (actually `rateup`) uses the `strftime(3)` `'%W'` option to format week numbers in the monthly graphs. The exact semantics of this format option vary between systems. If you find that the week numbers are wrong, and your system's `strftime(3)` routine supports it, you can try another format option. The POSIX `'%V'` option seems to correspond to a widely used week numbering convention. The week format character should be specified as a single letter; either `W`, `V`, or `U`.

Example:

Weekformat[ezwf]: V

## Two very special Target names

To save yourself some typing you can define a target called '^'. The text of every Keyword you define for this target will be PREPENDED to the corresponding Keyword of all the targets defined below this line. The same goes for a Target called '\$' but its text will be APPENDED.

Note that a space is inserted between the prepended text and the Keyword value, as well as between the Keyword value and the appended text. This works well for text-valued Keywords, but is not very useful for other Keywords. See the ``default" target description below.

The example will make mrtg use a common header and a common contact person in all the pages generated from targets defined later in this file.

Example:

```
PageTop[^]: <H1>NoWhere Unis Traffic Stats</H1><HR>
PageTop[$]: Contact Peter Norton if you have any questions<HR>
```

To remove the prepend/append value, specify an empty value, e.g.:

```
PageTop[^]:
PageTop[$]:
```

---

## And one more very special Target name

The target name '\_' specifies a default value for that Keyword. In the absence of explicit Keyword value, the prepended and the appended keyword value, the default value will be used.

Example:

```
YSize[_]: 150
Options[_]: growright,bits,nopercent
WithPeak[_]: ymw
Suppress[_]: y
MaxBytes[_]: 1250000
```

To remove the default value and return to the 'factory default', specify an empty value, e.g.:

```
YLegend[_]:
```

There can be several instances of setting the default/prepend/append values in the configuration file. The later setting replaces the previous one for the rest of the configuration file. The default/prepend/append values used for a given keyword/target pair are the ones that were in effect at the point in the configuration file where the target was mentioned for the first time.

Example:

```
MaxBytes[_]: 1250000
Target[myrouter.somplace.edu.2]: 2:public@myrouter.somplace.edu
MaxBytes[_]: 8000
Title[myrouter.somplace.edu.2]: Traffic Analysis for myrouter.somplace.edu IF 2
```

The default *MaxBytes* for the target myrouter.somplace.edu.2 in the above example will be 1250000, which was in effect where the target name myrouter.somplace.edu.2 first appeared in the config file.

---

## Some example mrtg.cfg files

---

### Minimal mrtg.cfg

```
WorkDir: /usr/tardis/pub/www/stats/mrtg
Target[r1]: 2:public@myrouter.somplace.edu
MaxBytes[r1]: 64000
Title[r1]: Traffic Analysis ISDN
PageTop[r1]: <H1>Stats for our ISDN Line</H1>
```

---

### Cfg for several Routers.

```
WorkDir: /usr/tardis/pub/www/stats/mrtg
Title[^]: Traffic Analysis for
PageTop[^]: <H1>Stats for
PageTop[$]: Contact The Chief if you notice anybody<HR>
MaxBytes[_]: 64000
Options[_]: growright
Title[isdn]: our ISDN Line
PageTop[isdn]: our ISDN Line</H1>
Target[isdn]: 2:public@router.somplace.edu

Title[backb]: our Campus Backbone
PageTop[backb]: our Campus Backbone</H1>
Target[backb]: 1:public@router.somplace.edu
MaxBytes[backb]: 1250000
# the following line removes the default prepend value
# defined above
Title[^]:
Title[isdn2]: Traffic for the Backup ISDN Line
PageTop[isdn2]: our ISDN Line</H1>
Target[isdn2]: 3:public@router.somplace.edu
```