

NAME

rrdcgi – Create web pages containing RRD graphs based on templates

SYNOPSIS

```
#!/path/to/rrdcgi [--filter|-f]
```

DESCRIPTION

rrdcgi is a sort of very limited script interpreter. Its purpose is to run as a cgi-program and parse a web page template containing special <RRD:: tags. **rrdcgi** will interpret and act according to these tags. In the end it will printout a web page including the necessary CGI headers.

rrdcgi parses the contents of the template in 3 steps. In each step it looks only for a subset of tags. This allows nesting of tags.

The argument parser uses the same semantics as you are used from your C-shell.

--filter|-f

Assume that rrdcgi is run as a filter and not as a cgi.

Keywords

RRD::CV *name*

Inserts the CGI variable of the given name.

RRD::CV::QUOTE *name*

Inserts the CGI variable of the given name but quotes it, ready for use as an argument in another RRD:: tag. So even when there are spaces in the value of the CGI variable it will still be considered to be one argument.

RRD::CV::PATH *name*

Inserts the CGI variable of the given name, quotes it and makes sure it starts neither with a '/' nor contains '..'. This is to make sure that no problematic pathnames can be introduced through the CGI interface.

RRD::GETENV *variable*

Get the value of an environment variable.

```
<RRD::GETENV REMOTE_USER>
```

might give you the name of the remote user given you are using some sort of access control on the directory.

RRD::GOODFOR *seconds*

Specify the number of seconds this page should remain valid. This will prompt the rrdcgi to output a Last-Modified, an Expire and if the number of seconds is *negative* a Refresh header.

RRD::INCLUDE *filename*

Include the contents of the specified file into the page returned from the cgi.

RRD::SETENV *variable value*

If you want to present your graphs in another time zone than your own, you could use

```
<RRD::SETENV TZ UTC>
```

to make sure everything is presented in Universal Time. Note that the values permitted to TZ depend on your OS.

RRD::SETVAR *variable value*

Analog to SETENV but for local variables.

RRD::GETVAR *variable*

Analog to GETENV but for local variables.

RRD::TIME::LAST *rrd-file strftime-format*

This gets replaced by the last modification time of the selected RRD. The time is *strftime*-formatted with the string specified in the second argument.

RRD::TIME::NOW *strftime-format*

This gets replaced by the current time of day. The time is *strftime*-formatted with the string specified in the argument.

Note that if you return : (colons) from your *strftime* format you may have to escape them using \ if the time is to be used as an argument to a GRAPH command.

RRD::TIME::STRFTIME *START/END start-spec end-spec strftime-format*

This gets replaced by a *strftime*-formatted time using the format *strftime-format* on either *start-spec* or *end-spec* depending on whether *START* or *END* is specified. Both *start-spec* and *end-spec* must be supplied as either could be relative to the other. This is intended to allow pretty titles on graphs with times that are easier for non RRDtool folks to figure out than “-2weeks”.

Note that again, if you return : (colon) from your *strftime* format, you may have to escape them using \ if the time is to be used as an argument to a GRAPH command.

RRD::GRAPH *rrdgraph arguments*

This tag creates the RRD graph defined by its argument and then is replaced by an appropriate tag referring to the graph. The **--lazy** option in RRD graph can be used to make sure that graphs are only regenerated when they are out of date. The arguments to the **RRD::GRAPH** tag work as described in the **rrdgraph** manual page.

Use the **--lazy** option in your RRD::GRAPH tags, to reduce the load on your server. This option makes sure that graphs are only regenerated when the old ones are out of date.

If you do not specify your own **--imginfo** format, the following will be used:

```
<IMG SRC="%s" WIDTH="%lu" HEIGHT="%lu">
```

Note that %s stands for the filename part of the graph generated, all directories given in the PNG file argument will get dropped.

RRD::PRINT *number*

If the preceding **RRD::GRAPH** tag contained and **PRINT** arguments, then you can access their output with this tag. The *number* argument refers to the number of the **PRINT** argument. This first **PRINT** has *number* 0.

RRD::INTERNAL <var>

This tag gets replaced by an internal var. Currently these vars are known: VERSION, COMPILETIME. These vars represent the compiled-in values.

EXAMPLE 1

The example below creates a web page with a single RRD graph.

```
#!/usr/local/bin/rrdcgi
<HTML>
<HEAD><TITLE>RRDCGI Demo</TITLE></HEAD>
<BODY>
<H1>RRDCGI Example Page</H1>
<P>
<RRD::GRAPH demo.png --lazy --title="Temperatures"
    DEF:cel=demo.rrd:exhaust:AVERAGE
    LINE2:cel#00a000:"D. Celsius">

</P>
</BODY>
</HTML>
```

EXAMPLE 2

This script is slightly more elaborate, it allows you to run it from a form which sets RRD_NAME. RRD_NAME is then used to select which RRD you want to use as source for your graph.

```
#!/usr/local/bin/rrdcgi
<HTML>
<HEAD><TITLE>RRDCGI Demo</TITLE></HEAD>
<BODY>
<H1>RRDCGI Example Page for <RRD::CV RRD_NAME></H1>
<H2>Selection</H2>
<FORM><INPUT NAME=RRD_NAME TYPE=RADIO VALUE=roomA> Room A,
      <INPUT NAME=RRD_NAME TYPE=RADIO VALUE=roomB> Room B.
      <INPUT TYPE=SUBMIT></FORM>
<H2>Graph</H2>
<P>
<RRD::GRAPH <RRD::CV::PATH RRD_NAME>.png --lazy
      --title "Temperatures for "<RRD::CV::QUOTE RRD_NAME>
      DEF:cel=<RRD::CV::PATH RRD_NAME>.rrd:exhaust:AVERAGE
      LINE2:cel#00a000:"D. Celsius">

</P>
</BODY>
</HTML>
```

EXAMPLE 3

This example shows how to handle the case where the RRD, graphs and cgi-bins are separate directories

```
#!/.../bin/rrdcgi
<HTML>
<HEAD><TITLE>RRDCGI Demo</TITLE></HEAD>
<BODY>
<H1>RRDCGI test Page</H1>
<RRD::GRAPH
  /.../web/pngs/testhvt.png
  --imginfo '<IMG SRC=/.../pngs/%s WIDTH=%lu HEIGHT=%lu >'
  --lazy --start -1d --end now
  DEF:http_src=/.../rrds/test.rrd:http_src:AVERAGE
  AREA:http_src#00ff00:http_src
>
</BODY>
</HTML>
```

Note 1: Replace `/.../` with the relevant directories

Note 2: The `SRC=/.../pngs` should be paths from the view of the webserver/browser

AUTHOR

Tobias Oetiker <tobi@oetiker.ch>