

# Howto configure Eddie/Elvin/eddiebrowser for data storing and graphing

Chris Miles

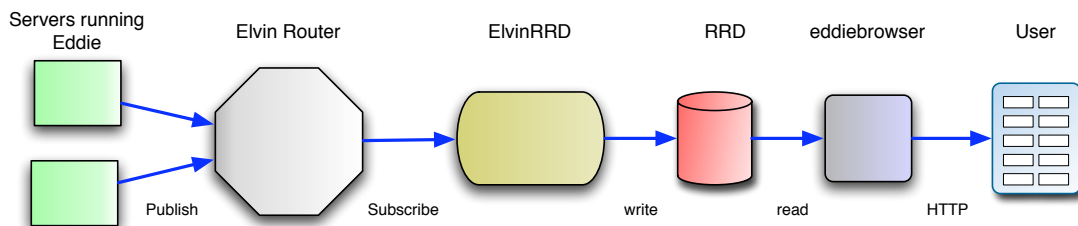
*Dec 21, 2005 [Ver 1.0]*

## 1 Introduction

The EDDIE Tool [1] can collect a lot of system data, but it is primarily a monitoring daemon that processes rules and actions, and so is not designed to store and display that data for trend analysis. For that, Eddie relies on other software components which are built for the job. One such solution gives Eddie the ability to publish system data to a networked messaging system, store that data in a central data store, and display that data graphically with a user friendly interface.

The software used to do this is a combination of: Elvin Router [2]; ElvinRRD [3]; RRDtool [4]; and eddiebrowser [5].

This document describes how to install and configure all the components to produce a networked system data collection, storage and graphing solution.



## 2 Eddie

Install Eddie using the docs that come with it. Make sure the basics are working. You will come back to it to add some configuration later.

## 3 Elvin

Install the Elvin Router Server:

- <http://www.mantara.com/products/elvin-router/>
- Edit `elvind.conf`, with at least:

---

```
protocol elvin:/tcp,none,xdr/0.0.0.0:2917
```

---

It should be possible to leave the rest of the config settings with default values. There are many configuration options as can be seen, for more advanced setups.

- Start `elvind`

- You can test it is working with:

---

```
$ telnet hostname 2917
```

---

and making sure you get a successful connection.

## 4 Elvin-Python Module

Install the Elvin-Python module (Elvin Python SDK):

- <http://www.mantara.com/products/elvin-sdk/python-sdk.php>
- You would normally install this with a command like:

---

```
$ sudo python setup.py install
```

---

- You can test it by running Python and trying

---

```
>>> import elvin
```

---

## 5 RRDtool

RRDTool & Python rrdtool module:

- <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
- Build and install rrdtool
- Make sure to build the Python rrdtool module that comes with rrdtool (it should be built and installed automatically if configure finds the Python interpreter).
- Test this by running Python and trying:

---

```
>>> import rrdtool
```

---

## 6 ElvinRRD

- <http://www.psychofx.com/elvinrrd/>
- Download and unpack elvinrrd into a directory.
- By default elvinrrd writes the RRD databases in `/export/rrd/` directory. If you want it elsewhere then edit `elvinrrd.cf` and globally modify all `"/export/rrd"` instances to your chosen directory..
- Create your rrd data directory. Example:

---

```
$ mkdir /export/rrd
```

---

- Make sure it can be written to by the user that will run the `elvinrrd.py` daemon
- Make sure `elvind` is running, then startup `elvinrrd.py` daemon with:

---

```
$ ./elvinrrd.py -d -c elvinrrd.cf -e elvin://localhost -l /var/log/elvin/elvinrrd.log
```

---

where “elvin://localhost” points to the address of the Elvin server; and “/var/log/elvin/elvinrrd.log” is the logfile to write to.

- Keep an eye on the log file for any errors.
- Testing: you can test ElvinRRD manually by using Python.

---

```
>>> import elvin
>>> e=elvin.connect('elvin://localhost')
>>> e.notify( ELVINRRD='loadavg1-test', loadavg1=1.0 )

$ tail /var/log/elvin/elvinrrd.log
2005-12-05 13:29:44 rrd.update( ('/export/rrd/loadavg/loadavg1-test.rrd', 'N:1.0') )
2005-12-05 13:29:44 Creating directory '/export/rrd/loadavg'
2005-12-05 13:29:44 Creating rrd: ('/export/rrd/loadavg/loadavg1-test.rrd', '-s', '60',
'DS:loadavg1:GAUGE:120:0:300', 'RRA:AVERAGE:0.5:1:1440', 'RRA:AVERAGE:0.5:10:1008',
'RRA:AVERAGE:0.5:60:720', 'RRA:AVERAGE:0.5:480:1095')

$ ls -l /export/rrd/loadavg/
total 70
-rw-r--r--  1 elvin  elvin      35248 Dec  5 13:29 loadavg1-test.rrd

$ /opt/rrdtool/bin/rrdtool info /export/rrd/loadavg/loadavg1-test.rrd
filename = "/export/rrd/loadavg/loadavg1-test.rrd"
rrd_version = "0003"
step = 60
last_update = 1133789384
ds[loadavg1].type = "GAUGE"
ds[loadavg1].minimal_heartbeat = 120
[...]
```

---

## 7 Eddie Config

It is now time to configure Eddie to automatically send system statistics to ElvinRRD via the Elvin router.

- Edit eddie.cf and configure the ELVINURL setting:

---

```
ELVINURL='elvin://localhost'
```

---

- Now edit a rules file (eg: common.rules) and add this directive:

---

```
SYS loadavg_rrd:
    rule='1'          # always true
    scanperiod='1m'  # sample every 1 minute
    action=elvinrrd('loadavg1-(h)s', 'loadavg1=%(loadavg1)f'),elvinrrd('loadavg5-%(h)s',
'loadavg5=%(loadavg5)f'),elvinrrd('loadavg15-%(h)s', 'loadavg15=%(loadavg15)f')
```

---

- “%(h)s” will be substituted with the hostname of the server.
- “%(loadavg1)f” will be substituted with the loadavg1 value (a floating point number)
- the elvinrrd() action is a built-in Eddie action to publish messages to the Elvin network. The messages are formatted in such a way that ElvinRRD will receive them. The action is called 3 times, one for each RRD.

- Fire up Eddie and make sure there are no errors (check the logfile) and check the ElvinRRD logfile to see if the loadavg1 message was received. A new RRD file should have been created for it (named with the hostname of the server) as well as the loadavg5 and loadavg15 data files.

## 8 eddiebrowser

- <http://eddie-tool.net/eddiebrowser/>
- Untar and copy somewhere to install it. Example:

---

```
$ sudo mkdir /opt/eddiebrowser
$ cd eddiebrowser-0.6.1
$ sudo cp -r * /opt/eddiebrowser
```

---

- Create data directory and configure eddiebrowser. Example:

---

```
$ sudo mkdir /export/rrd
$ sudo vim /opt/eddiebrowser/eddiebrowser.cgi
GLOBAL_CONFIG = '/opt/eddiebrowser/configs/eddiebrowser.cfg'
$ sudo vim /opt/eddiebrowser/configs/eddiebrowser.cfg
rrd_dir /export/rrd
rrd_conf_dir /opt/eddiebrowser/configs
dummy_rrd /opt/eddiebrowser/dummy.rrd
```

---

- Create a symbolic link in your cgi-bin directory to eddiebrowser.cgi. Example:

---

```
$ cd /var/www/cgi-bin
$ sudo ln -s /opt/eddiebrowser/eddiebrowser.cgi
```

---

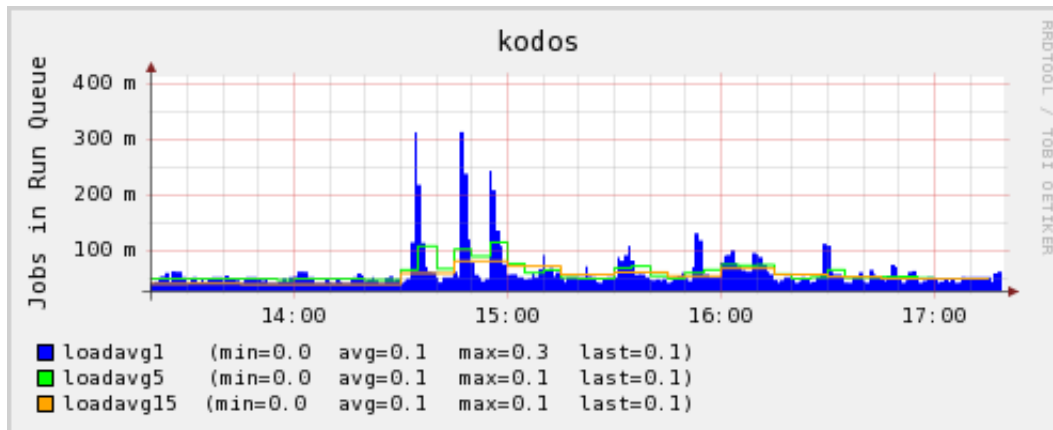
- Make sure your web server is configured to follow symlinks.
- Now point your web browser to the location of the cgi. Example:

---

```
http://localhost/cgi-bin/eddiebrowser.cgi
```

---

- If you see the control panel with some hosts listed in the pull-down then that is a good start.
- Now select a host and click “go”. If you see a graph then pat yourself on the back. An example is shown below.
- If you see an image containing an empty graph with an error message underneath it, then correct the error (the web server error log file should contain a copy of the error message, which may be easier to read).
- If you see a broken image icon instead of a graph, you will have to examine the web server error log file and see if it contains an error message. If not, try copying the URL of the image itself, fetch that URL using a command line tool (such as wget or curl) and examine the result. The result may contain an error or traceback explaining what the problem is.



## References

- [1] <http://eddie-tool.net/>
- [2] <http://elvin.dstc.edu.au/>
- [3] <http://www.psychofx.com/elvinrrd/>
- [4] <http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/>
- [5] <http://eddie-tool.net/eddiebrowser/>