I've just bought a new house, and having a geeky interest in home automation and a more serious interest in climate change I bought some lightwaveRF home automation kit including an electricity monitor.

I work in IT and am familiar with the Cacti graphing tool which provides a front end for rrdtool a graphing program specifically designed for displaying time-series data. This seemed like the serious interest in climate change, I wanted to try out some home automation with a view to first measuring our energy usage and then hopefully be able to take some action to reduce it.

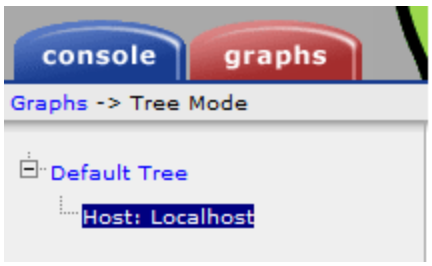
I've just bought the LightwaveRF Wifi link and energy meter. Annoyingly, once I logged into the LightwaveRF site, I found merely a teaser about a "concierge" service to show historical power usage. I don't want to wait until September, and since it sounds like there might be a cost associated with this service when it does arrive, I took it upon myself to try to make my own graphing system.ideal solution since I have a Linux box on all the time (home server, media streaming and automation). So I took it upon myself to build a cacti plugin for the LightwaveRF wifi link and energy monitor.

I started with this useful blog post which got me started with PHP for LightwaveRF automation, and then using the LightwaveRF open API to work out how to receive the response, I followed the instructions here about building a PHP cacti plugin. I built a php script able to send a UDP datagram requesting current energy data and then receive and parse the result and pass it back to cacti.

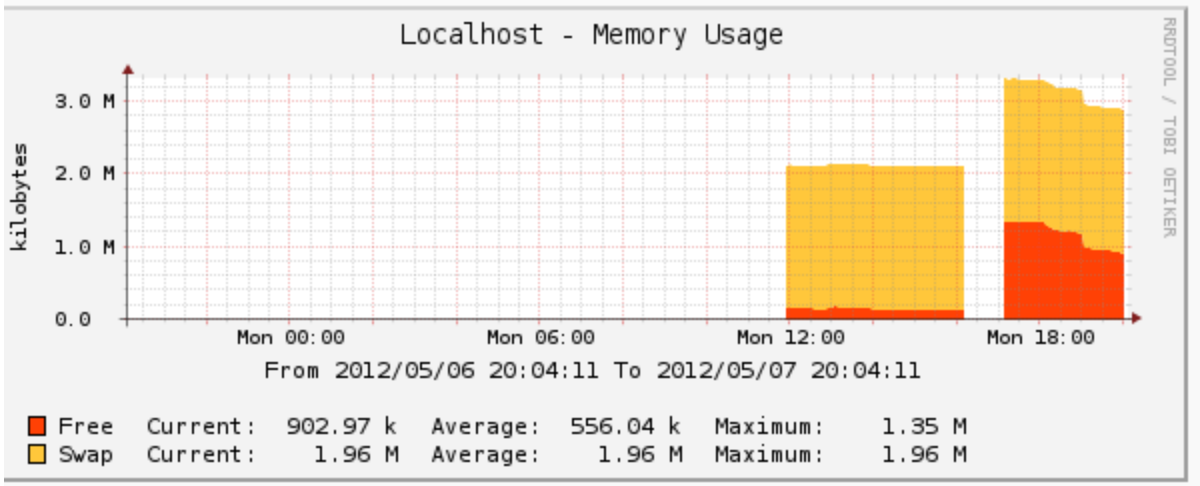
Unfortunately, Cacti have changed their PHP script method since that post was written so I had to follow the cacti docs to port my script into their new format. In the end however I was able to create a working script which could be called by Cacti to retrieve energy consumption data from a LightwaveRF Wifi link with energy meter.

The script is available from Github here:
git://github.com/stephenjirvine/lightwaverf_energy_poller.git

In order to use this script you will need a *working* cacti installation, I use ubuntu and was able to install using the ubuntu package manager apt-get. Once installed, you log in, reset your password and away you go. To help you make sure your graphs are working Cacti includes a set of 4 graphs you should be able to see by clicking the Graphs button in the top left.



You should now see some graphs below that look like this:



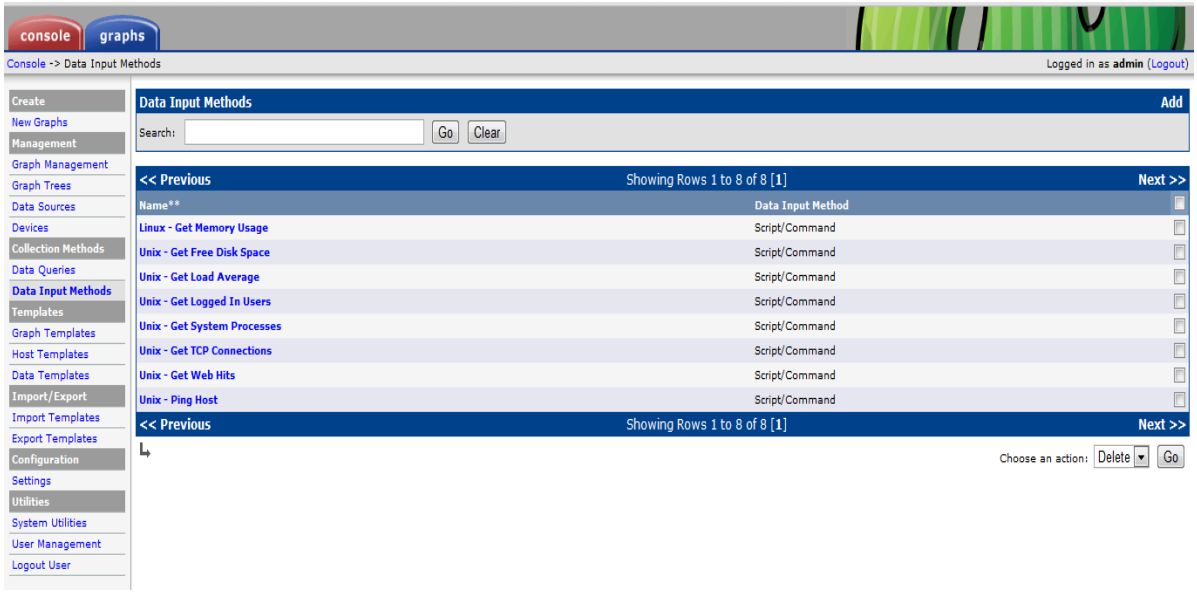
These are the cacti builtin graphs so if you can't see them you need to fix cacti before going any further.

Hint: check the user your apache server is running under has permission to access the directories where cacti stores it's data.

Once you've verified you can see the builtin graphs, copy the php script up to a folder on the cacti computer that is accessible to the cacti process, (I chose the scripts folder in the cacti directory itself /use/share/cacti/site/scripts, but you can put it anywhere).

Important: the file must be called ss_lwrfenergypoller.php, this is a cacti convention.

Once done you need to create a data input method for LightwaveRF, so click Console at the top and click on Data Input method. Then click add in the top right to make a new one to add to the existing list.



Fill in the fields as above, if you want to change the name that's fine but the Input String must read:

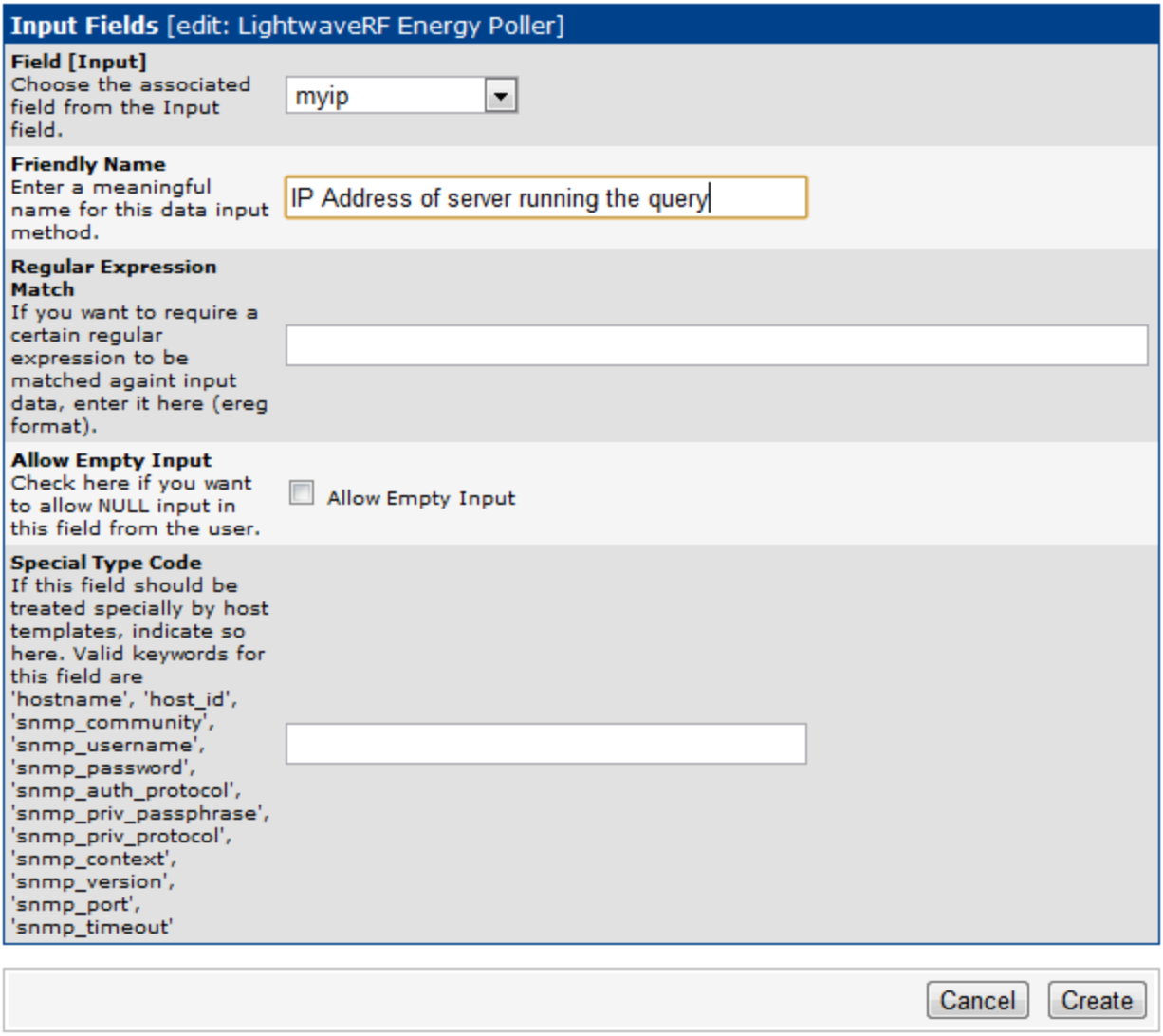
/usr/share/cacti/site/scripts/ss_lwrfenergypoller.php ss_lwrfenergypoller <myip> <wifilinkip> <send_port> <recv_port> <broadcast_string>

If you put your script in a different place then make sure your path goes above.

Click create and then click Add in the Input Fields section:

You will need to do this 5 times, one for each of the fields in <> brackets in the Input string above, what we're doing here is defining the arguments we're going to supply to the energy poller script.

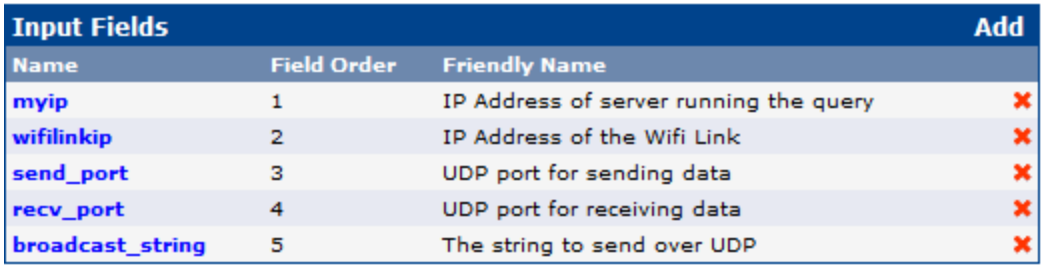
Select the first Input field "myip" fill in the details as shown here:



Then click create, do this for each of the input fields:

myip = IP Address of the computer running cacti
wifilinkip = IP Address of the Wifi Link
send_port = UDP port for sending data
recv_port = UDP port for receiving data
broadcast_string = The string to send over UDP

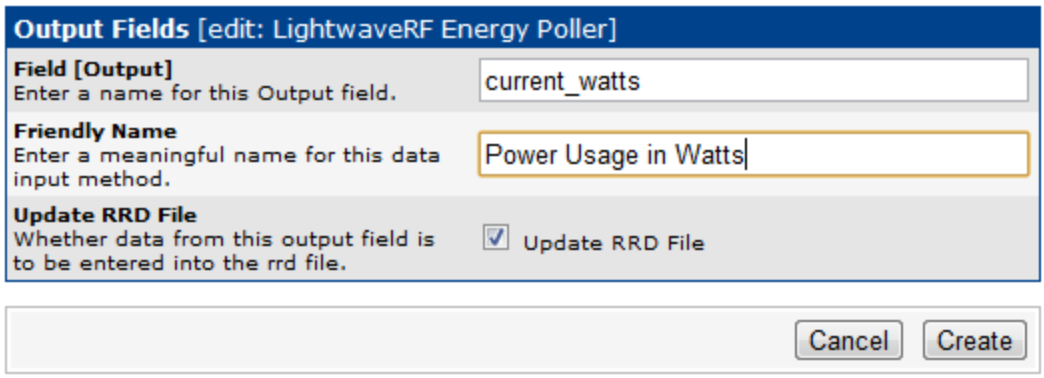
By now your input fields table should look like this:



Input Fields			Add
Name	Field Order	Friendly Name	
myip	1	IP Address of server running the query	✖
wifilinkip	2	IP Address of the Wifi Link	✖
send_port	3	UDP port for sending data	✖
recv_port	4	UDP port for receiving data	✖
broadcast_string	5	The string to send over UDP	✖

I decided to break out these fields as input arguments to the script to allow the same script to be re-used in the event of LightwaveRF adding other pollable devices later. We shouldn't need to edit the script in the future, just call it with different inputs.

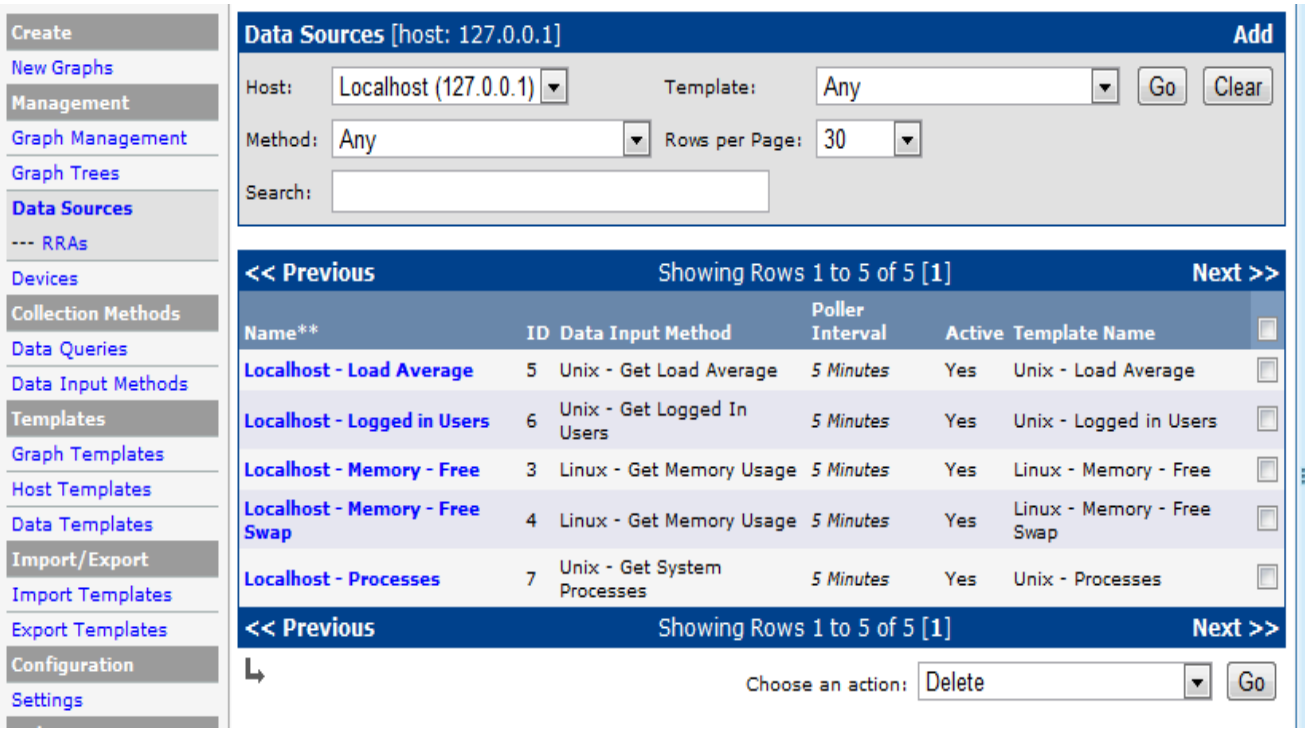
Now we need to define the output of the script. Click add in output fields, this should be quicker, there's only one output, the current power measured in Watts:



Hit create and you should see Save Successful. in the top.

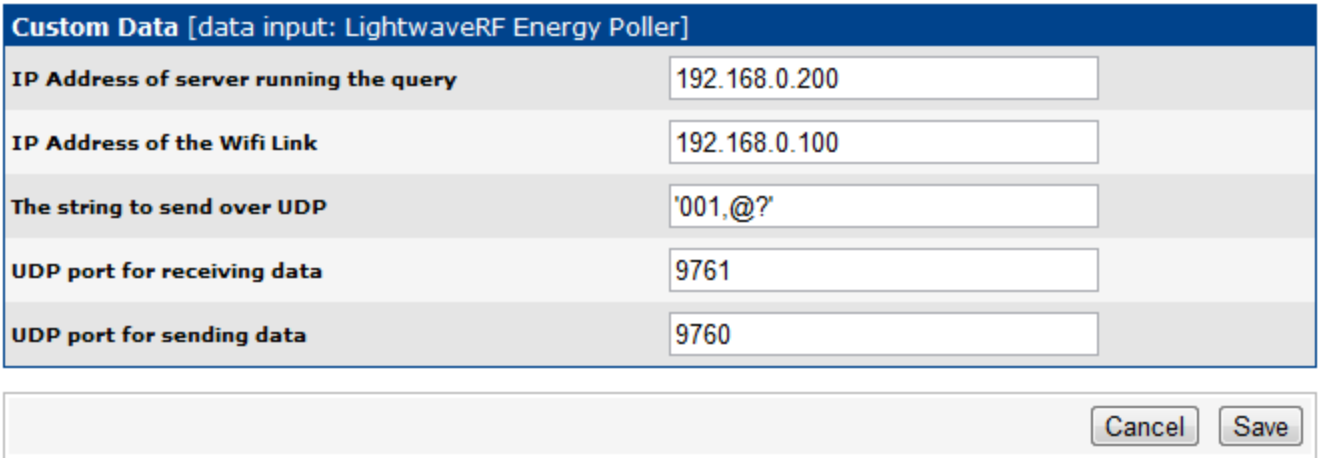
You've successfully defined the method by which we collect data from the WiFi link, now we have to set up a Data Source to actually do it.

On the menu on the left click Data Sources, then add in the top right:



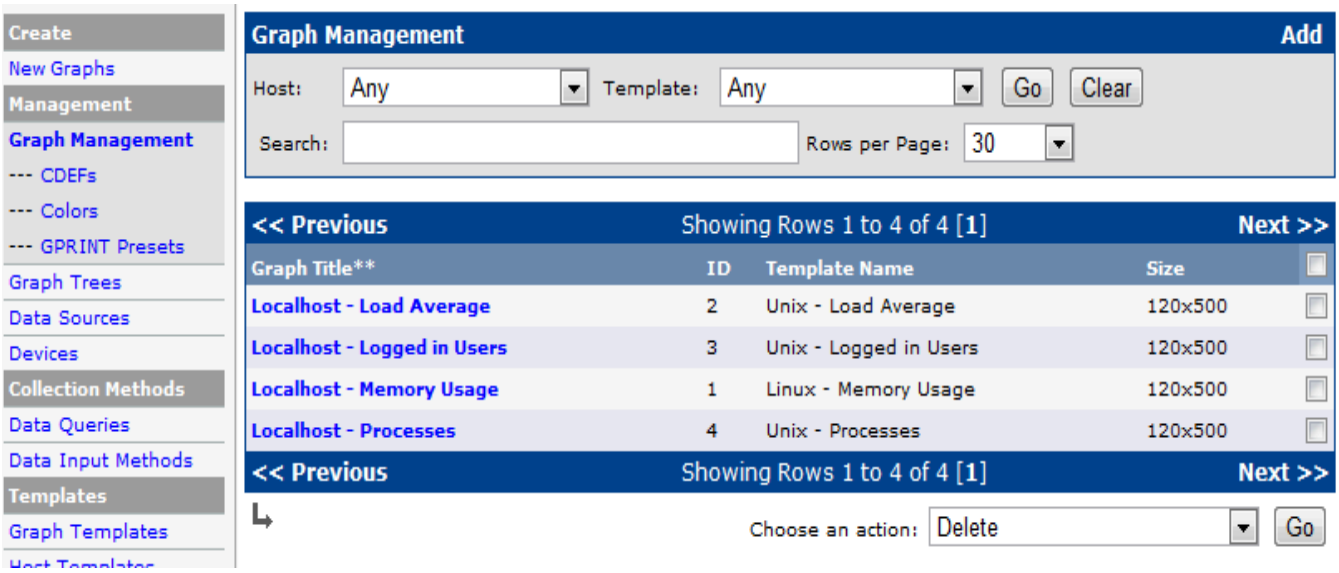
We don't have a Template to work from, and we should leave this data source associated with Localhost, click create and fill the rest in as below. Leave all the RRAs selected. Hit Create.

Another box should pop out of the bottom, now you can populate the inputs we defined earlier:

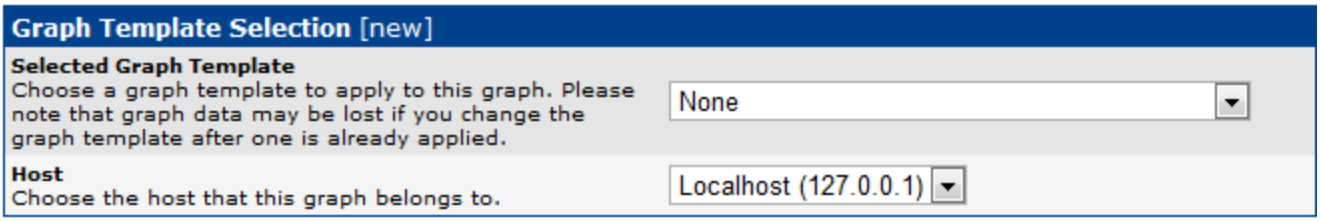


Please note these just order themselves alphabetically based on what's in the description string, so be careful what you put where. You will obviously need to change the IP Address values to match your environment.

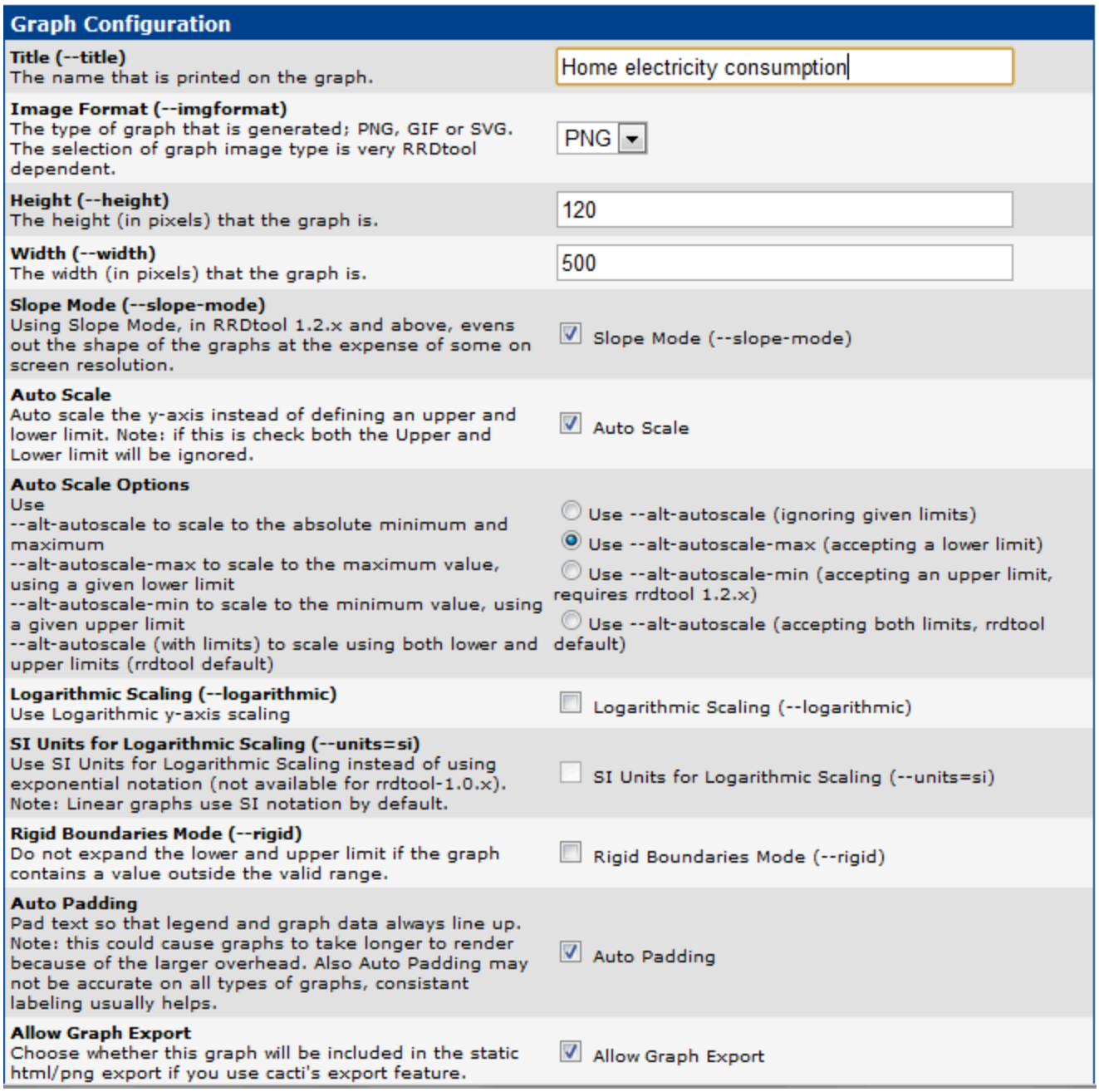
Once you save this, you should be collecting data, but now we need to define a graph to view it, on the left go to Graph Management., and click add in the top left.



Leave template blank and assign this graph to localhost:



Then I just gave it a name and left everything else as it was.



Once you save this dialogue, a graph designer pops up above it:

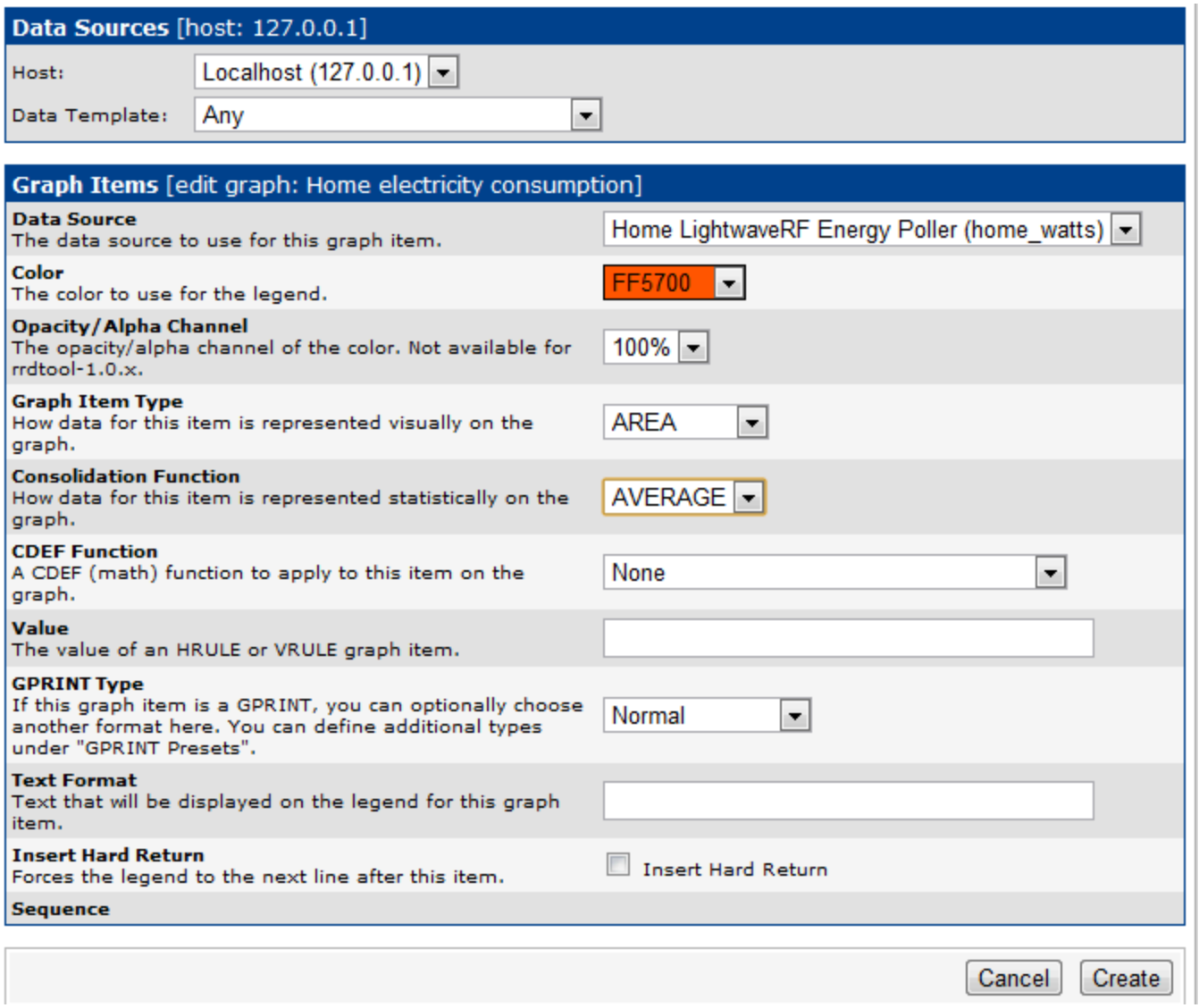
Graph Items [edit: Home electricity consumption]				Add
Graph Item	Data Source	Graph Item Type	CF Type	Item Color
No Items				

RRDTool Command:

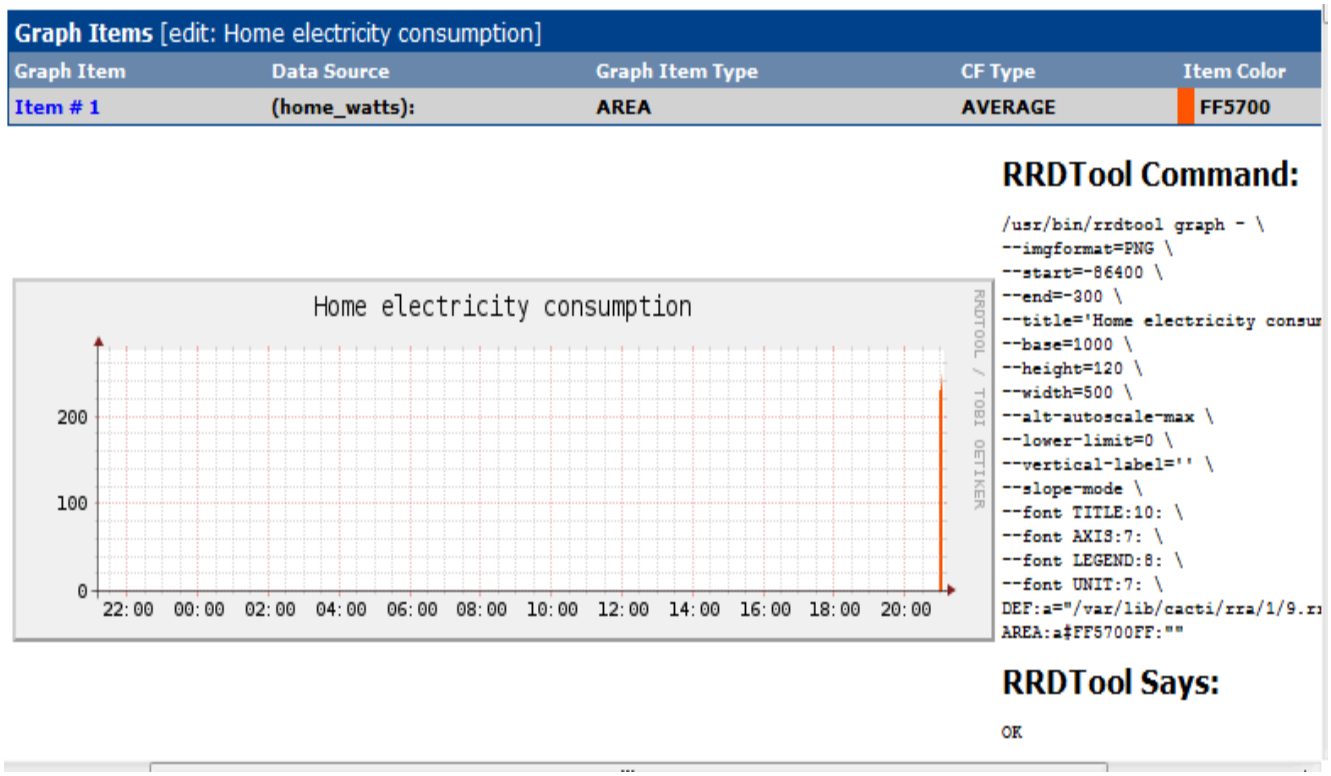
```
/usr/bin/rrdtool graph - \
--imgformat=PNG \
--start=-86400 \
--end=-300 \
--title='Home electricity consumption' \
--base=1000 \
--height=120 \
--width=500 \
--alt-autoscale-max \
--lower-limit=0 \
--vertical-label='' \
--slope-mode \
--font TITLE:10: \
--font AXIS:7: \
--font LEGEND:8: \
--font UNIT:7: \
```

RRDTool Says:

Use the add dialogue to add items to the graph, here we add an area graph of our energy usage, in a nice orange colour:

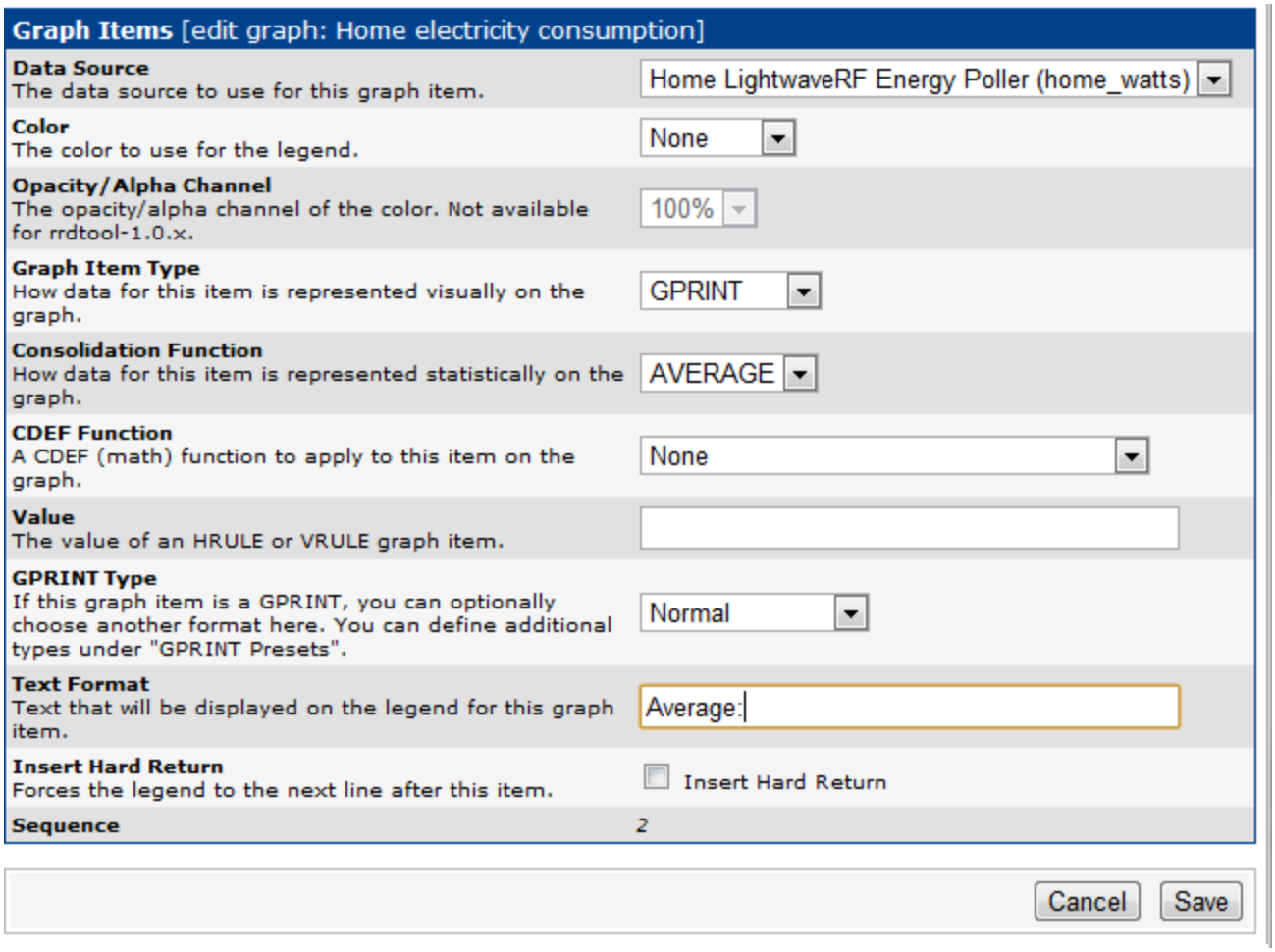


We should see our new graph appear below:

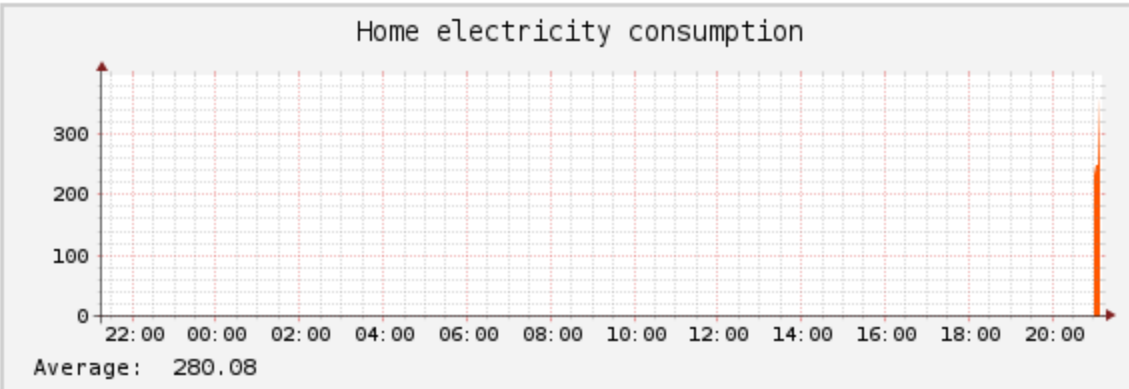


Ace! It worked. Now we see some data appearing in our first graph.

We can add other items, for example a gprint function to show Average power usage:

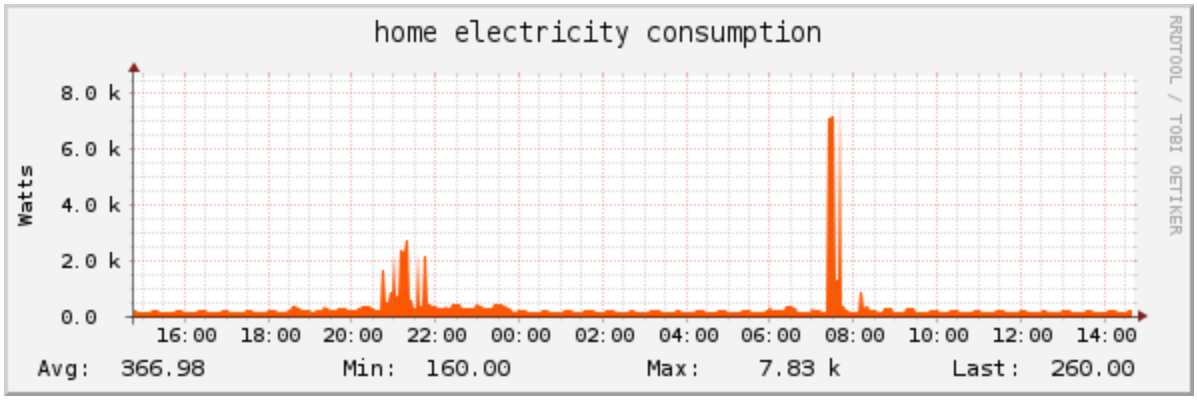


Now our graph has an average calculated on the fly and added to the bottom left:



You should have a play around here with the different options and labels, there's a lot you can do.

In the end I added a few more GPRINT statements to show min, max etc etc and after a day's logging here's my output.



The 2kw spike around 21:00 is us cooking dinner and the big 7.5kw spike just before 08:00 is morning showers.