Automatically Keeping Your Google Sitemap up to Date

Many CMS's will automagically generate a Google Sitemap, but if you are not using a CMS, or it doesn't have that functionality then the hard work is left to the order. This guide will show you how to configure a Linux based server to automatically generate a Google sitemap at regular intervals. The system will then check whether anything new has been added to the sitemap, and if it has it will ping Google to notify them of the change.

First you need to grab Google's sitemap generator from

https://www.google.com/webmasters/tools/docs/en/sitemap-generator.html

Follow that guide to configure the Generator itself, I manually update a file called urllist.txt whenever I update the site, so that was my primary focus. However I have also configured the generator to configure the access logs just in case I have missed anything (the added benefit is that if someone has gained access to your system, a quick review of the new sitemap will show any documents that they have published without your knowledge.)

Right, the next thing to do is to create the script that will check if any changes have been made, so create a BASH script called sitemap_updated.sh

```
#!/bin/bash
#
# Small Script to submit Sitemap Updates to Google
#
# Full, URL Encoded address of Sitemap
ADDRESS="http%3A%2F%2Fbenscomputer.no-ip.org%2Fsitemap.xml"
# Local Path to sitemap
PATHXML=/path/to/directory/containing/sitemap
#Filename of sitemap
FILENAME="sitemap.xml"
# Google request page
PINGAD="www.google.com/webmasters/tools/ping?sitemap="
FQADDY="$PINGAD""$ADDRESS"

# OK has the sitemap changed?

diff -wBa "/tmp/$FILENAME" "$PATHXML""$FILENAME" > /dev/null
if [ "$?" == "1" ]
then
# The sitemap has changed
# Notify Google
wget --delete-after $FQADDY
# grab an up to date copy of sitemap for next run
cp "$PATHXML""$FILENAME" /tmp/$FILENAME
else
# Sitemap hasn't changed.
echo "Sitemap unchanged, exiting"
exit
fi
```

Save this script somewhere memorable, then make it executable

chmod +x sitemap_updated.sh

Now that both utilities are present on the server, we need to get them to run regularly. So add them to your Cron Jobs, making sure that the sitemap_updated.sh script will run sometime after the generator.

So my Crontab now reads

@hourly /home/ben/programs/sitemap_gen-1.4/sitemap_gen.py --config=/home/ben/programs/sitemap_gen-1.4/config.xml --testing

*20,44 0,7-23 * * * /home/ben/programs/sitemap_gen-1.4/sitemap_updated.sh*

As I am unlikely to have updated the site after about 12 I haven't scheduled the sitemap_updated.sh script to run in the silent hours. This will obviously depend on your patterns, and you may want to put one scheduled job in the midst of the silent hours just to be safe.

Now, before the thing will run, you need to create a file for diff to examine for the first time. So assuming your sitemap is called sitemap.xml run the following command

echo "Blank File" > /tmp/sitemap.xml

Otherwise the system will error out, and Google will not be notified of any changes.

Now you're set up and ready to go, all you have to do when you add a new page is update urllist.txt with the URL of the page you have added (make sure it is a fully qualified web address), so the file should be in the format

http://benscomputer.no-ip.org/
http://benscomputer.no-ip.org/Archives.html

And so on.

To automatically generate a HTML sitemap, download HTML_Sitemap_generator from http://benscomputer.no-ip.org/projects.html#SitemapGen