Creating a Google Sitemap

The packaged php scripts allow creating a Google sitemap according to the rules defined on
http://www.google.com/schemas/sitemap/0.84/ . There are two files in the package:

> sitemap.php
> sitemap_init.php

The files were tested using version 5.2.0 of php.

sitemap.php

This file is the main php script to generate the sitemap. It contains the class definition of the sitemap class as well as the constructor and callback functions.
The file includes the sitemap_init.php file, if it exists.

sitemap_init.php

This file contains two parts: Variable assignment for configuration parameters and a function called back from sitemap.php to generate the sitemap.

How To

Sitemap.php can be executed in two modes; run on the server to generate the sitemap of a site on a deployed server, or it is called from the command line to generate a sitemap from a local directory structure.

Output is by default sent to the standard output, ie web server, using echo commands. Additionally the output can be sent to a user defined file, assuming that file open permissions allow this operation.

Main features

By default the script generates a sitemap of all files found from '.', same directory the sitemap.php script is located following all branches of subdirectories. By default all files in all sub-trees are included.
By default the Priority field of the Google sitemap is set to 0.5 (a relative value between 0 and 1) and the re-visit frequency is set to *monthly*. See the xml scheme on http://www.google.com/schemas/sitemap/0.84/ for more details on the valid fields.
Some additional features of the script are:

Definition of Root directory

By default the script starts reading files and following the directory structure at ".". Setting the variable $smroot in sitemap_init.php to another location changes the default sitemap root.

Definition of Directory separator

By default the script uses the system defined variable DIRECTORY_SEPARATOR to separate components in the file path written to the sitemap output. This works if the script is run on a system where the global variable is set properly (I found sites where this was not the case). If the script is run on a system for example on a Windows development system and the deployment system, where you expect Google to find the sitemap, is not windows, but a system using the forward slash '/' as directory separator you should set the variable $d_separator in sitemap_init.php to '/'.

Definition of deployment URL Root

If you develop a site in one place, but you like to generate the sitemap for the deployed site you can set the deployment URL by assigning a valid URL to the variable $deployedPath in sitemap_init.php.
$deployedPath = http://www.merope.com sets the deployed path of a local web.

Definition of generated sitemap file

By default the sitemap is sent to standard out using echo. You can also write the sitemap to a file. Just set the variable $smfilename in sitemap_init.php to the name of the file.

Exclusion of directories

The script traverses directories recursively from the root. If you don't want all directories included in the sitemap you can include the directory name, relative to the root directory in the ignore array:

```
$ignoreArray[] = '/testing';
$ignoreArray[] = '/private/testing';
```

Will exclude all files in testing and in any subdirectory of `/testing` and all files and directories below `/private/testing`.

Exclusion of files by name

You can add files to be ignored from the sitemap by adding the file to the array ignoreFiles.

```
$ignoreFiles[] = 'ignore_this.html';
```

excludes the file 'ignore_this.html' from the sitemap. Note if a file with the same name is found in different directories all occurrences will be excluded.

Exclusion of file by suffix

If you want to exclude a list of files with a specified suffix from the sitemap, just add it to the ignoreExtensions array:

```
$ignoreExtensions[] = '.jpg';
$ignoreExtensions[] = '.js';
```

excludes all jpeg and JavaScript files.

Definition of Frequency per file

To overwrite the default sitemap frequency as defined in the Google XML model you can set the frequency on a per file basis.

```
$fileChangeFreq["index.html"] = "weekly";
$fileChangeFreq["news.html"] = "hourly";
```

Changes frequency of index.html to weekly and of news.html to hourly. Valid frequency values according to the Google sitemap definition are:

```
always   (dynamically generated)
hourly
daily
weekly
monthly
yearly
never    (archived pages)
```

Definition of Priority per file

To overwrite the default sitemap file priority as defined in the Google XML model you can set the priority on a per file basis.

```
$filePriority["index.html"] = "1.0";
$filePriority["vacations.html"] = "0.7";
$filePriority["work.html"] = "0.1";
```

Priorities are in the range of 0.0 to 1.0 and, according to Google, just set relative priorities of the pages.

he following modifiers are called from the sitemap callback function `setlocalSitemapDefaults` after the sitemap class been initiated[1]. Assume the variable `$mysitemap`,

```
$mysitemap = new RJGoogleSiteMap(…),
```

contains the newly created object.

Exclusion of FrontPage derived files

After the sitemap object is created, you can ignore all FrontPage generated files by calling.

```
$mysitemap->excludeFrontpage();
```

This excludes the contents of the four FrontPage directories:

---

[1] Note this is done automatically in the sitemap.php script after having included the sitemap_init.php file.

```
        _derived, _vti_cnf, _vti_pvt, _private
```

Definition of default Frequency

Calling
```
    $mysitemap->setDefaultFrequency("weekly");
```
sets the default frequency for all files not explicitly specified on a per file basis to `weekly`.

Definition of default Priority

Calling
```
    $mysitemap->setDefaultPriority("0.5");
```
sets the default priority for all files not explicitly specified on a per file basis to 0.5.

Support for robots.txt file

Calling
```
    $mysitemap->considerRobotsTxt("robots.txt");
```
Opens and parses the file given as argument, default `robots.txt` and uses the `Disallow` tags in a robots file to exclude files and directories according to the rules defined on
```
    http://www.robotstxt.org/wc/norobots.html.
```

Callback function

If the function
```
    setlocalSitemapDefaults($sitemap, $smf);
```
exists, usually defined in sitemap_init.php it is called with the parameters
```
    $sitemap, the sitemap object
```
and
```
    $smf, the name of the default output file as specified in the
    global $smfilename.
```

Sitemap Generation

To generate a sitemap, even more than once, you can call
```
    $mysitemap->processAndWriteXML($outputfile, $echotoo);
```
This will generate the sitemap, write it to file `$outputfile` and if the second parameter is set to `TRUE` to standard out, using `echo`.
You can generate the sitemap more than once using different output file settings and you also can use the modifier functions
```
    setDefaultFrequency
    setDefaultPriority
    considerRobotsTxt and
    excludeFrontpage
```
to change default behavior.
For example
```
    $mysitemap->processAndWriteXML("sitemapwithfp.xml", FALSE);
    $mysitemap->excludeFrontpage();
    $mysitemap->processAndWriteXML("sitemapwithoutfp.xml", TRUE);
```
writes a sitemap with FrontPage files to `sitemapwithfp.xml` and a sitemap without FrontPage files to `sitemapwithoutfp.xml` as well as to standard out (echo).

Version History

Version 1 by RJ Softwares (www.rjsoftwares.com)
Version 2, Feb 2007, by Peter Pircher, Merope Consulting, www.merope.com – see comments in php for changes.

Version history and evolution of the script by different authors explains the different styles (global definition array for object instantiation vs. modifiers on instantiated object.
All this is for usage as is, no warranties, guarantees and just let us (RJ Softwares and Merope Consulting) know if you use it and enhance it.