

"Advanced" Apache Headers/Footers

Scott Gasch

scott.gasch@gmail.com

Copyright © 2007 Scott Gasch

This article describes how to set up headers and footers around your webpages using apache and `mod_layout`. It also covers using `mod_setenvif` with server-side includes to make dynamic headers without having to write (and run) a bunch of scripts.

This document is available in several formats:

- One large HTML file: <http://wannabe.guru.org/scott/hobbies/howtos/apache.html>
(<http://wannabe.guru.org/scott/hobbies/howtos/apache.html>)
- Plain (7 bit ASCII) text: <http://wannabe.guru.org/scott/hobbies/howtos/apache.txt>
(<http://wannabe.guru.org/scott/hobbies/howtos/apache.txt>)
- Adobe PostScript: <http://wannabe.guru.org/scott/hobbies/howtos/apache.ps>
(<http://wannabe.guru.org/scott/hobbies/howtos/apache.ps>)
- Adobe Portable Document Format (PDF): <http://wannabe.guru.org/scott/hobbies/howtos/apache.pdf>
(<http://wannabe.guru.org/scott/hobbies/howtos/apache.pdf>)
- Microsoft Rich Text Format (RTF): <http://wannabe.guru.org/scott/hobbies/howtos/apache.rtf>
(<http://wannabe.guru.org/scott/hobbies/howtos/apache.rtf>)

1. Introduction

It seems like a lot of websites (including this one) wrap their HTML pages in a header and footer. It looks nice, unifies the site, etc...

A header is a good place for the page title and some navigation aides like breadcrumbs or a site menu. A footer is a good place for stuff like a stock copyright notice or a "last modified" timestamp.

I probably don't have to tell you that most people don't hardcode the header and footer into every page on their site. Most people are too lazy for this... and besides, if you followed this route then when you inevitably wanted to change some text in the header you would have to edit every file to make the update.

I don't know how other people do the header/footer trick but I found a cool little Apache plugin called `mod_layout` (http://tangent.org/?view=mod_layout) that works great.

2. Setting Up `mod_layout`

Setting up `mod_layout` is pretty easy. There are already some good examples on Tangent Software's website (http://tangent.org?view=mod_layout) but I will include some examples here for completeness.

The first thing you need to do is make sure that your `httpd.conf` file is telling Apache to load the `mod_layout` module:

```
LoadModule layout_module      libexec/apache22/mod_layout.so
```

If adding this causes your webserver to refuse to load then you probably need to build `mod_layout` which is easy and will be covered later on in this guide.

Once you have the module loaded you can use directives in your `httpd.conf` file to tell it what to do. Here are some examples from my own site configuration.

This first one tells `mod_layout` not to try to add a header or footer on certain filenames:

```
LayoutIgnoreURI *.rss
LayoutIgnoreURI *.xml
LayoutIgnoreURI *.raw
LayoutIgnoreURI *.htm
```

Now that we've gotten that out of the way, let's tell it what headers and footers to actually add:

```
<Directory "/usr/local/www/data/scott">
  Options +Indexes
  AllowOverride All
  AddOutputFilter LAYOUT html htm shtml
  LayoutComment On
  LayoutHeader /wrappers/scott-header.html
  LayoutFooter /wrappers/scott-footer.html
</Directory>
```

Well, pretty straightforward. In directory such-and-such, I want all the files with matching extensions to be wrapped by this header and that footer. The bit about "LayoutComment" simply instructs `mod_layout` to put HTML comments around the header and footer text.

3. Dynamic Headers

Well, now you have some boilerplate header and footer on your whole website. Congratulations.

What's that? You don't want the same header on every page? And some footers should have different information too?

You have a few choices. First, you can separate your site into several directories and use the recipe above to add a different header and footer on a per-directory basis. This works great if you want all the files in the same directory to have the same headers and footers.

I didn't want this though. I needed every file to potentially have a different header and footer. So I started looking for another way to do this "dynamic header" trick. The first way that I came up with was to use a CGI script for the header. Simply change the path of your `/wrappers/scott-header.html` to something like `/cgi-bin/scott-dynamic-header.pl`, write the script, and it will work. The "script" I'm talking about here is a CGI script... If you're lost at this point better stop reading and look for a tutorial about what CGI scripts are because the details of CGI programming are beyond the scope of this guide. Or read on because my final solution doesn't involve any CGI.

CGI is well and good but it's slow, puts a load on the server, and requires (very basic) programming skills. My main objection is the "load on the server" thing. What happens if your site gets slashdotted (not likely, I admit, but it might happen)? Will your server hold up? I have a hint: webservers can handle heavier loads if they aren't spinning up a bunch of CGI processes all the time. One process per page load is a bit excessive.

But how, then, do we get the "dynamic" behavior of a CGI powered header/footer? Well, we could use server-side includes to do some of the work. (Again, if you don't know what this is go google it). But the variables available to a SSI page are limited. Or are they?

```
<Directory "/usr/local/www/data/scott">
  Options +Indexes
  AllowOverride All
  AddOutputFilter LAYOUT html htm shtml
  LayoutComment On

  SetEnvIf Request_URI "scott" HEADER_IMAGE=/scott/images/misc_top.gif
  SetEnvIf Request_URI "scott\/*$" HEADER_IMAGE=/scott/images/scott_top.gif

  SetEnvIf Request_URI "photos" HEADER_IMAGE=/scott/images/photos_top.gif
  SetEnvIf Request_URI "info" HEADER_IMAGE=/scott/images/info_top.gif
  SetEnvIf Request_URI "hobbies" HEADER_IMAGE=/scott/images/hobbies_top.gif
  SetEnvIf Request_URI "tools" HEADER_IMAGE=/scott/images/tools_top.gif

  SetEnvIf Request_URI "books" HEADER_IMAGE=/scott/images/books_top.gif
  SetEnvIf Request_URI "chess" HEADER_IMAGE=/scott/images/chess_top.gif
  SetEnvIf Request_URI "contact" HEADER_IMAGE=/scott/images/contact_top.gif
  SetEnvIf Request_URI "hiking" HEADER_IMAGE=/scott/images/hiking_top.gif
  SetEnvIf Request_URI "house" HEADER_IMAGE=/scott/images/house_top.gif
  SetEnvIf Request_URI "location" HEADER_IMAGE=/scott/images/location_top.gif
  SetEnvIf Request_URI "music" HEADER_IMAGE=/scott/images/music_top.gif
  SetEnvIf Request_URI "photos" HEADER_IMAGE=/scott/images/photos_top.gif
  SetEnvIf Request_URI "skiing" HEADER_IMAGE=/scott/images/skiing_top.gif
  LayoutHeader /wrappers/scott-header.shtml
  LayoutFooter /wrappers/scott-footer.shtml
</Directory>
```

This is the same chunk of my configuration file with some additions. The `SetEnvIf` blocks are used to set the state of an environment variable (`HEADER_IMAGE`) based on the URI that the browser requested. This variable is then available to the SSI header. It just does something like this:

```
<IMG SRC="<!--#echo var="HEADER_IMAGE" -->">
```

All I wanted to do "dynamically" with my headers was change the image at the top. While I could have done this with a bunch of static HTML headers, this is easier and pretty fast. You may want to do more sophisticated things... but I think this recipe would probably extend nicely. Note that to use this you'll need to enable `mod_setenvif` in your Apache `httpd.conf` configuration.

4. Code Hacking

That section title sounds so cool, doesn't it. Well, don't be too impressed. At least reserve your admiration until you see the dirty hack in question.

At this point I have my nice dynamic headers using `mod_layout`, server side includes, and `mod_setenvif` in a supporting role. What more could I want?

Well I wanted to do a little message on the footer that said when the page was last modified. My first attempt was to use the environment variable `LAST_MODIFIED`, which is provided to server side include pages. Unfortunately this doesn't work because it ends up being the timestamp when you last modified the footer page, not the page being wrapped.

If you read the `mod_layout` site carefully you might have noticed that it reports to set a different environment variable for you called `LAYOUT_LAST_MODIFIED`. Don't believe it. Maybe it did at some point but the code in the latest version I have is commented out:

```
// apr_table_setn(subr->subprocess_env, "LAYOUT_LAST_MODIFIED",
// apr_ht_time(r->pool, r->finfo.st_mtime, cfg->time_format, 0));
```

Why is it commented out? Good question. I see nothing dangerous about this line of code so I tried uncommenting it and rebuilding `mod_layout`. Suffice to say it fails. It seems like someone changed the API in the Apache `apr` stuff and now this doesn't work anymore (which is probably why it's commented out).

Here's my dirty hack:

```
char buf[30];
apr_rfc822_date(buf, r->finfo.mtime);
apr_table_setn(subr->subprocess_env, "LAYOUT_LAST_MODIFIED", buf);
```

It's dirty for a couple of reasons but maybe not what you're thinking. The 30 char buffer is large enough to handle any return from `apr_rfc822_date` so no, it's not a buffer overflow. But when you are writing a server side include you can set some variable called `timefmt` (or something like that) which is supposed to cause times and dates to be displayed in any way you want (see `strftime(3)`) I couldn't find a function that would do this for me in the `apr` so I settled for this one (which blatantly ignores the contents of `timefmt`). Also the 30 should be something more like `APR_RFC822_DATE_LEN` but that would mean including an extra header file.

I figured that while I was in the code I might as well create another environment variable to do something I wanted: store the short name of the file being wrapped. It turns out that there is already one that stores the long name (i.e. full URI) of the page being wrapped and it's called `LAYOUT_FILENAME`. Unlike `LAYOUT_LAST_MODIFIED`, this one works. But it returns something like `"/path/to/your/filename.html"` and all I want is `filename.html`. Without further ado, here's the code:

```
char *p = strdup(r->filename);
```

```
if (p) {
    char *q = strrchr(p, '/');
    if (q) {
        apr_table_setn(subr->subprocess_env, "LAYOUT_SHORT_FILENAME", q + 1);
    }
    free(p);
}
```

5. Conclusion

Well, I hope you now have a spiffy site with headers and footers around your pages where you want them. Feel free to mail me (<mailto:scott.gasch@gmail.com>) if you want to tell me about errors on this page or share some better way to do the same thing.