

- [Home](#)
- [Blog](#)
- [Resources](#)
- [About](#)
- [Contact](#)
- [Feeds](#)

The Definitive Post on Gzipping your CSS

Published in [CSS](#) on Sunday, June 13th, 2004

This post is the source for the most definitive/recent/tested method (in the scope of this blog) of gzipping your CSS when either Apache compression module is not available for use.

[7-Zip Download. Free](#)

download Free 7zip software. Free Full Version,
no need to buy.
[freeze.com](#)

[Download Audio Converter](#)

Compress MP3 WMA WAV AAC M4A AMR
Convert All Audio Files Fast!
[www.avs4you.com/AVS-Audio-Converter](#)

[Mediven Compression Sale](#)

Save 20-50%, Huge Selection 800-982-0939 -
Fast, Free Shipping
[www.Discountsurgical.com/Mediven](#)

[Improve ASP Performance](#)

Increase server speed and delivery with
advanced ASP code optimization
[www.Codekko.com](#)

Ads by Google

Note: This post was originally published on June 13th, 2004. It is being republished today because: the topic has been discussed in a few places recently, and, well, I accidentally republished it when editing it.

Background Info

This whole idea started back in February of 2004 and has evolved through several iterations thanks to feedback from different users.

Because of these iterations, pieces of this concept are strewn about several posts on this blog (see [history](#), below). *This post* is being written with the objective of holding the most recent and up-to-date version for gzipping your CSS. My apologies for the poor management of this topic. Lesson learned.

Why and how?

CSS files for larger sites can become pretty large themselves. Gzipping or compressing these files has shown to provide a reduction in the neighborhood of 70-80% of the original file size, a fairly significant 'weight loss'.

The obvious method to accomplish compression (on an Apache server) is to use either the `mod_gzip` or `mod_deflate` module ([more here](#)), these modules are not available to everyone. If you have access to those modules, follow the given link, otherwise... PHP provides an alternative compression method (two actually) and this is exactly what we'll leverage here - so one could think of this method as a 'poor man's `mod_gzip`'.

Outline

There are two different, equally effective methods that can be used.

The first method involves adding a little snippet of PHP to the top of your CSS file, and then renaming your CSS file with a 'php' extension. The second, cleaner and more elegant solution involves adding two little files to your CSS directory, one of them an `.htaccess` file, the other a PHP file that contains the same snippet of code used in the first method.

The snippet

A little bit of PHP magic is needed to make this happen. The following code is all it takes:

```
<?php
ob_start ("ob_gzhandler");
header("Content-type: text/css; charset: UTF-8");
header("Cache-Control: must-revalidate");
$offset = 60 * 60 ;
$ExpStr = "Expires: " .
gmdate("D, d M Y H:i:s",
time() + $offset) . " GMT";
header($ExpStr);
?>
```

This code does 4 things:

1. It uses PHP's [ob_gzhandler](#) to send compressed data. This function will first check to see if the browser requesting the file will accept 'gzip,deflate' encoding; if not it sends the file uncompressed.
2. It sends a header for the content type and character set for the file - in this case text/css and UTF-8.
3. The next step sends a 'cache-control http header'. Here 'must-revalidate' ensures that any information that you pass along about the freshness of your document is obeyed.
4. The final step is to send an 'Expires' header, to set an age on how long our cached file will last. Here we set it to expire in one hour.

Method One

This method, though not as clean as the second, is just as effective. All you need to do is place the PHP snippet from above into the top of your CSS document. Then, rename your CSS file with a 'php' extension, and then refer to that file when linking your css file, for example:

```
@import url(mycss.php);
```

Method Two

Method two is cleaner and more elegant, as it does not require the addition of any extra code to your CSS file, nor do you have to change the extension of the file. Clean and simple.

Two steps are necessary to implement this method.

Step one

First, you want to save the snippet provided above in a file called 'gzip-css.php' to the directory that contains your CSS files.

Step two

Here, simply add the following to an .htaccess file and save the file to the same directory as your CSS files:

```
AddHandler application/x-httpd-php .css  
php_value auto_prepend_file gzip-css.php  
php_flag zlib.output_compression On
```

This code does 3 things:

1. The first line tells Apache to send all .css files to the PHP script handler.
2. The second line prepends the code snippet to your CSS file.
3. **Optional:** The third line tells PHP to use its built-in negotiated output compression automatically for every page it parses. If you use this method for compression, there is no need for having `ob_start ("ob_gzhandler");` in the code snippet. So choose one or method or the other (see [note](#)).

Thanks to [David](#) for noting that this will prepend the file to all files found in the directory being used, so best to keep only CSS files in this directory.

Some notes

Confirmation

It is recommended that you check to see that this method is in fact working for you. Some people have reported that method one works fine while method two doesn't.

As with anything, do your homework! Test locally first, on a testing server, and be sure to examine the log files of your server to ensure that this is working!

Compression

PHP offers two types of compression, you can make an informed decision on which one to use by reading up on them: [ob_gzhandler](#), [zlib compression](#).

Caching

While the CSS files are explicitly set to cache, in our tests, Opera doesn't seem to cache the files when compressed.

Parsing

Some people expressed concern about invoking PHP's parser for this, wondering if it would either tax the server or result in no overall speed benefit.

With respect to the first concern, as the CSS files are cached, the file need only be processed on the first visit. As such, we have not observed this to be much of a concern.

With respect to the second point, we've seen 70-80% reduction in the delivered file size. So for a 10K CSS file, your looking at a reduction of 7.5kb, for example. Though this may seem like much, it is significant - would you be willing to add another 7.5kb to your header graphic for no reason at all? Neither would we!

History

For reference, here is a list of past posts related to this topic:

- [Discussing CSS Management and Optimization](#)
- [Applied CSS Management and Optimization](#)
- [Http Compression - Results and user numbers](#)
- [Organizing and Optimizing Your CSS - Some ideas](#)
- [More Notes on Gzipping CSS](#)
- [Gzipping your CSS with PHP](#)

Thanks

A thank you should be extended to all those who gave feedback and helped to extend and get this to the point that it is, in particular [Alun Bestor](#) and Robbert Broersma.

Thanks to [Lars Kasper](#) for catching the missing charset in the Content-type header.

Feedback

If you have an idea, comment or suggestion pertinent to this topic, please [let us know via our contact form](#).

Comments and Feedback

[Mike P.](#)

[Tue, 15th of June, 2004](#)

I realize after that fact that referring to this as the **definitive** post on gzipping CSS may sound (more than) a little presumptuous. Please understand that I was speaking in relation to the scope of this blog.

[Mike P.](#)

[Tue, 15th of June, 2004](#)

A couple of bits that need mentioning:

1. Early versions of Netscape 4 reportedly would send the correct headers to accept compressed content but lacked the ability to handle it. Therefore it is not recommended to send compressed files to that browser.
2. In Andy King's book, [Speed Up Your Site](#), it is mentioned that "webmasters have found that browsers inconsistently decompress .css files". We have yet to come across any browser trouble with compressed CSS files. If you happen to run into trouble, please [let us know through our contact form](#).

[Mike P.](#)

[Mon, 27th of September, 2004](#)

Thanks to [Matt](#) for pointing something out to me:

```
$offset = 60 * 60 ;
```

In the section on headers above, that value sets the CSS file to cache for 1 hour. If you aren't changing your site's CSS very often, it's a good idea to set that for a longer value.

If you do change your CSS and are worried about users not getting the new version, you could simply set a new filename for your CSS file.

[Sven Fischer](#)

[Thu, 29th of March, 2007](#)

Hi,

verry thanks for the interesting article and the good php code snippet - I need it ;-)

Greetings from Germany

Sven

David Rodger

[Thu, 29th of March, 2007](#)

It's probably worth mentioning that method 2 will autoprepend the file for any request to a file in that directory. So it's advisable to place only CSS in the directory so that only CSS files are handled in this manner.

David Rodger

[Thu, 29th of March, 2007](#)

Correction:

method 2 will autoprepend the file for a request to any file in that directory which is handled by PHP.

[Mike](#)

[Thu, 29th of March, 2007](#)

Thanks for catching that David. I always stick css in `/assets/css/`, so that was assumed to be the case :-)

Daniel F

[Fri, 30th of March, 2007](#)

Would these methods work for compressing javascript as well?

[Mike](#)

[Fri, 30th of March, 2007](#)

Hi Daniel, I think they will, but you must send the correct headers. At the time that I wrote this originally (2004), there were reports that browsers were having trouble with gzipped JS.

Cal Henderson wrote about [this here](#) (see heading *Compression*), and it looks like he's suggesting that gzip compression of JS isn't yet a good idea.

[Respiro the logo designer](#)

[Mon, 2nd of April, 2007](#)

Thanks for republishing the article. These discussions about CSS compression are really welcome!

[Jon Henshaw](#)

[Wed, 4th of April, 2007](#)

It seems that the best solution would be to use a server that supports gzip for all text based files. But if you can't get on a server that supports that, then your solution certainly seems like the next best thing.

[John](#)

[Thu, 5th of April, 2007](#)

I had an interesting browser experience with this format for css.

I had accidentally deleted the *snippet* from the top of my CSS file and uploaded it as a `css.php` file.

Firefox refused to render the page while Safari, IE7 and Opera 9 just carried on as though nothing had happened.

Lesson learned.. When using the WebDeveloper extension for live CSS editing, it will not save the file

with the *code snippet* intact. :)

[masterev](#)

[Mon, 23rd of February, 2009](#)

Thanks for reigniting this article. It is very well written and something i will attempt to incorporate across a range of my sites.

I do have one question though and please consider me a noob to gzipping.

When i gzip my css file which works perfectly using method 2 i can no longer view images referenced in the css file. Is there a way around this?

[louis](#)

[Mon, 20th of July, 2009](#)

Thanks for the great article!

Talal Ebdah

[Wed, 22nd of July, 2009](#)

Hi\nDoes this method work on JavaScript as well?

If yes, could anyone tell me how, please?

Thanks

Talal Ebdah

[Wed, 22nd of July, 2009](#)

Sorry about that. but I have just modified the code to work on javascript as well.

following the second method you need to modify the second line of the snippet as the following:\nheader(

[Home](#) » [Blog](#) » [Web Development](#) » [CSS](#)

Check out the blog categories for older content

- [9 Rules](#)
- [Application Programming Interfaces](#)
- [Blogging](#)
- [Contingency Design](#)
- [CSS](#)
- [Fiftyfourteen](#)
- [General](#)
- [Link Wrap](#)
- [Marketing](#)
- [News, Rants and Ephemera](#)
- [Programming and Scripts](#)
- [Search Engines](#)

- [Web Development](#)
- [Widgets that Work](#)
- [Working on the Web](#)

The latest from my personal website, [Mike Papageorge.com](http://MikePapageorge.com)



Sitepoint's [web development books](#) have helped me out on many occasions both for finding a quick solution to a problem but also to level out my knowlegde in weaker areas (JavaScript, I'm looking at you!). I am recommending the following titles from my bookshelf:

The Principles Of Successful Freelancing

I started freelancing by diving in head first and getting on with it. Many years and a lot of experience later I was still able to take away some gems from this book, and there are plenty I wish I had thought of beforehand. If you are new to freelancing and have a lot of questions (or maybe don't know what questions to ask!) do yourself a favor and at least check out the sample chapters.

- [More Info](#)
- [Download Free Sample Chapters](#)

The Art & Science Of JavaScript

The author line-up for this book says it all. 7 excellent developers show you how to get your JavaScript coding up to speed with 7 chapters of great theory, code and examples. Metaprogramming with JavaScript (chapter 5 from Dan Webb) really helped me iron out some things I was missing about JavaScript. That said each chapter really helped me to develop my JavaScript skills beyond simple Ajax calls and html insertion with libs like JQuery.

- [More Info](#)
- [Download Free Sample Chapters](#)

The PHP Anthology: 101 Essential Tips, Tricks & Hacks

Like the other books listed here, this provides a great reference for the PHP developer looking to have the right answers from the right people at their fingertips. I tend to pull this off the shelf when I need to delve into new territory and usually find a workable solution to keep development moving. This only needs to happen once and you recoup the price of the book in time saved from having to develop the solution or find the right pattern for getting the job done..

- [More Info](#)
- [Download Free Sample Chapters](#)

- [Home](#)
- [Blog](#)

- [Resources](#)
- [About](#)
- [Contact](#)
- [Feeds](#)

Search

Copyright © 2001 - 2010 - Fiftyfoureleven.com - Web Design Resources and Blog