

Improving website performance

Website performance is becoming a bit of a lost art. As more and more people get broadband, and that broadband gets quicker, then website owners spend less time looking at performance. After all, when you have a quick connection, why bother?

What these people forgot is that not everybody is on the same super-quick broadband. But, even if that's not the case, we all know that we prefer a quick website over something slower. Then factor in the fact that many people have hosts that limit bandwidth, and the need for efficiencies becomes more apparent.

In this article, I'm going to be using my own websites for demonstration purposes, so I will be talking about HTML (or rather XHTML), PHP and MySQL. I've provided details in [a past post](#) about useful tools for reducing images in size, so don't intend to cover this again.

The aim is simple... to look at performance and reduce loading times and bandwidth.

So, where to start.

Well, there are various tweaks we can do to Apache servers and PHP and MySQL installs to improve such things but, if like me, you're getting hosting from a third party you're unlikely to be able to do anything about this, so I'll be concentrating on coding. However, changes to site configuration files – such as .htaccess – are usually allowed so I will include these.

The first thing to do, and excuse me if this sounds obvious, is to try and identify where the slowdown occurs. Is it the host server (possibly PHP or MySQL scripts) or client side (HTML, CSS, JavaScript, Images, etc.)?

When it comes to code, there are 3 basic rules on how we can improve this...

1. **Compress** (or, as it's called elsewhere, "minify") so that the files are smaller
2. **Reduce** by removing redundant code
3. **Improve** code so that it works more efficiently

PHP

How we code our PHP can make a difference in the speed. Here is a useful list of tips on how to code more efficiently in PHP...

<http://reinholdweber.com/?p=3>

This is a pretty good list and is pretty much all I can say on the subject, other than – and this applies to all other coding – to look through your code and try and remove redundancy.

Improving website performance

MySQL

The general advise is..

1. Do as little SQL processing as possible. Reduce down what the SELECT fetches, if you can
2. One of the biggest things is select *. If you don't need all the columns, only ask for the ones you need. If you don't need all the rows, ask for the ones you need, but ask in a way that exploits the indexes already there.
3. Use Indexes wherever possible

And whilst I'm at it, don't forget to look at SQL Injection. It won't help with speed, but is essential for site security.

HTML/XHTML

Other than some of the tips mentioned previously (reduce code where appropriate, etc), there is no magic bullet for this one. Compression programs give very little benefit (reducing code by a couple of percent). There are some that make huge improvements but these come with decompression routines and can slow down the display of pages, as they are decompressed browser-side.

Additionally, many compression routines will strip out characters that will then mean the page can't be validated without errors.

JavaScript

Unlike HTML/XHTML, I've found compression of JavaScript to be beneficial. My particular favourite is the one provided [on the Creativyst website](#). The [Litebox](#) script, for example, will compress to 67% of its original size, saving a third of the total size. Unlike other sites and tools, I've never found the Creativyst version to break a script.

CSS

Like JavaScript, compression can help with CSS. Be wary, though, of various "hacks" that are used in CSS for cross-browser compatibility. Compression may break these.

Therefore, I put any such "hacks" in a separate CSS file, which I don't then compress.

My recommendation is the [online compressor at CSSdrive](#), which can provide some impressive results (a 28% saving on the CSS that I tried). Using this compresses the main stylesheet for the BMTG website down to 72% of its original size – a saving of 28% on pretty much each page.

I'd recommend you set the following options...

Improving website performance

Spaces and tabs

- Replace multiple spaces with just one space
- Remove space around chars ;:}{,
- Leave space between selectors (?)
- Leave space between properties (?)
- Remove tabs

New lines handling

- Leave lines as they are
- Replace multiple empty lines with just one empty line
- Remove all newlines

Comment handling (?)

- Don't strip any comments
- Strip ALL comments
- Strip comments at least chars long

Other options

- Compress color codes where possible (?)
- Remove unnecessary semi-colons
- One command per line (?)

Having said that, I've achieved even greater compression using manual search/replaces (but doing it for each stylesheet every time I change them is laborious). One day I might write a program to do it for me!

.HTACCESS

There are two options, which can be set via your .htaccess file, which will help further.

1. [GZIP](#) – assuming your host server has GZIP installed, this will activate this compression functionality
2. [Far future expires](#) – this adds an “expires” header to your files

Both of these options should be used carefully and fully tested. The latter option, `Far_future_expires`, should be of particular note. This caches JS and CSS the first time a user visits. It's only then an advantage when they return again. Bear in mind that if you then change of these files you'll need to give them a different name, so document control will be useful here.

Both of these, plus a minify option, is also available via a script named `PHP_SPEEDY`. I will be covering this in a second article, however.

Server Calls

The number of calls you make to the server for files to make up a resultant web page can have a big impact on load time. Combining images, stylesheets, etc, is the best solution.

Now, you can do this manually, or there's a rather useful and clever script for automating this on [the takaz blog](#).

Improving website performance

Yahoo! Exceptional Performance

Yahoo! Have a section of their Developer Network titled [Exceptional Performance](#). It states that their “team evangelizes best practices for improving web performance”.

First of all, I’d recommend their [list of performance rules](#). This includes full explanations of each recommendation.

Secondly, and to back this up, they’ve introduced a Firefox plugin that will check a website for performance issues, using their aforementioned rules. This is named [Yslow](#) and requires the [Firebug](#) developer plugin as well.

Now, some of these recommendations are going to be for large-scale commercial sites, including having a number of interspersed host servers, etc., so I suspect it’s doubtful that anyone with a small scale site is going to be able to achieve a high rating.