# Lickity Split (/blog)

## Zoompf's Web Performance Blog

### Note: Archived Content

This is the archived version of the Zoompf blog. Since our acquisition by Rigor (/blog/2015/10/zoompf-gets-acquired), all our new research and posts on web performance are being published on The Rigor Blog (http://rigor.com/blog)

# Check These 3 Things Before You Deploy

Zoompf Performance (/contact) on April 10, 2015. Category: Optimization (/blog/category/optimization)

(http://pixabay.com/en/sunset-silhouettes-military-705296/)

*Image courtesy of Pixabay.*

There are a number of steps to complete prior to deploying any web software into production, and every development shop or department has (or should have) its own very thorough process. But outside of that process, we've found these three very important performance items that are often overlooked. Any software development team would do well to check these three things before deploying production software.

## 1. Combining CSS and JS Files

Combining CSS and JS improves performance by reducing the number of files, and thus the number of requests the browser is required to make to load the page. More files means more requests, and this issue is only magnified on mobile devices. It is common practice to break your CSS and JS into lots of little files to better organize them. That's ok if you did that for development purposes, but not for production deployment. Leaving all these files as individual files causes your audience's browsers (mobile and desktop) to make a new HTTP request for each file. Do you have a system in to combine those together again?  Minify (http://code.google.com/p/minify/) (php), Sprockets (http://getsprockets.com/) (Ruby), Jawr (https://jawr.dev.java.net/) (Java), and YUI (http://yuicompressor.codeplex.com/) (.Net) are just a few of the available tools that you can utilize to make sure your CSS and Javascript files are properly combined before production deployment.

## 2. File Versioning

Adding expiring headers causes the page load time to decrease (meaning a faster page load) for subsequent page views/return visits from repeat visitors. However, you can use far future caching only if you have a way to change the URL when the content changes.  Do you have a system in which the URL for each file will change when the contents of that file change? If not, you should consider including an embedded date/time stamp or build number in your URLs, so the URLs change with each build. Many developers overlook this performance gaining technique.

## 3. Performance Testing

From each release of your software to the next release, your team may make a few changes or lot of changes to your code. But a change is still a change. How do you know if performance has improved or been compromised with each change to your code? One simple method of determining performance changes *before* production deployment is to do an audit using free tools like WebPageTest.org (http://WebPageTest.org/), and Zoompf's free report (/free) and see if new problems have been introduced. When you test not only for bugs, but for performance (/blog/2015/03/treat-web-performance-issues-like-software-bugs), you may find that new issues impact whether you should release the software into production.

## Next Steps

Performance is something you need to think about during development, rather than after your software is already in production. In order to keep performance top of mind during development, rather than only after deployment, you must have a plan to develop for performance. These 3 things are a good start on your path to build fast websites. If you are interested in building in performance, you will love Zoompf. We have a number of free tools that help you detect and

correct front-end performance issues on your website: check out our free report (/free) to analyze your website for common performance problems, and if you like the results consider signing up for our free alerts (/alerts) to get notified when changes to your site slow down your performance.

---

Next Post (/blog/2015/04/how-to-keep-your-site-fast-for-mobile-friendly)

Earlier Post (/blog/2015/04/new-features-multiple-interactive-waterfalls-and-har-importing)

**Comments**

Have some thoughts, a comment, or some feedback? Talk to us on Twitter @zoompf (https://twitter.com/zoompf) or use our contact us form (/contact).

## Zoompf Becomes Rigor Optimization!

Zoompf's web performance product is now Rigor Optimization. Learn more (http://rigor.com/features).

## Get Your Free Report

Get a free detailed performance analysis of your website right now.

**Free Performance Report (http://rigor.com/free)**

## Blog Topics

announcement (/blog/tag/announcement) apache (/blog/tag/apache) asp.net (/blog/tag/asp-net) best practice (/blog/tag/best-practice) bug (/blog/tag/bug) caching (/blog/tag/caching) CDN (/blog/tag/cdn) Chrome (/blog/tag/chrome) compression (/blog/tag/compression) conference (/blog/tag/conference) CSS (/blog/tag/css) DevOps (/blog/tag/devops) Free (/blog/tag/free) Google (/blog/tag/google) How Fast Is? (/blog/tag/how-fast-is) HTML (/blog/tag/html) HTTP (/blog/tag/http) humor (/blog/tag/humor) IE (/blog/tag/ie) images (/blog/tag/images) JavaScript (/blog/tag/javascript) Lose The Wait (/blog/tag/lose-the-wait) Lossless Optimization (/blog/tag/lossless-optimization) Microsoft (/blog/tag/microsoft) performance (/blog/tag/performance) PHP (/blog/tag/php) PNG (/blog/tag/png) Presentations (/blog/tag/presentations) recommendation (/blog/tag/recommendation) report (/blog/tag/report) Safari (/blog/tag/safari) SEO (/blog/tag/seo) SPDY (/blog/tag/spdy) SSL (/blog/tag/ssl) tools (/blog/tag/tools) video (/blog/tag/video) webp (/blog/tag/webp)