79 TIPS AND TRICKS FOR MAGENTO PERFORMANCE IMPROVEMENT

✓ A Checklist to Faster Load Times and Higher Conversion Rates

Web2Market

Your website visitors crave fast load times and speedy product search. In fact, almost half of all web users expect a site to load in under two seconds. Any more than that and they start hitting their browsers' back buttons at alarming rates. When it comes to page response and load times, one measly second can mean the difference between conversion and abandonment.

That single second may also determine whether you make it to the first page of Google's search rankings, or end up buried on page three.

Simply put: a faster site means higher sales, more positive reviews, and greater customer loyalty.

Magento®

While Magento has empowered countless businesses with its incredible flexibility and ease of use, it's notoriously slow. Dismal load speeds are one of the top user complaints. The good news is, you can give your slow webstore a jolt with a few tricks, and a little expert guidance.

If you need help implementing these tips and tricks for your e-commerce site, connect with us on Twitter @Web2Market and tag your question with #AskMagentoExpert.

Think of us as your Phone-a-Friend option on "Who Wants to Be a Magento Millionaire?" Only you can call on us as many times as you want.

How To Speed Your Way Through the Checklist

Step 1: Print this checklist and keep it handy so you can work through every optimization trick at your own pace.

Step 2: Perform our 10 quick DIY configurations for a warm-up. Follow the step-by-step instructions to see results right away.

Step 3: Optimize with or without our expert assistance. The remaining 49 Application Level Fixes, 5 Configuration Level Fixes, and 14 Server Level Fixes can be knocked out on your own, or with the guidance of a Web2Market Magento expert.

Step 4: Note your Impact Potential. Each tip is ranked for its potential to impact your site's performance. You can use the rankings to help you prioritize changes. However, the Tips and Tricks work best together; a single change--even if it's High Impact--is not likely to make a noticeable difference. Your site will start to speed up after completing several fixes, and will keep gaining velocity as you check your way through the entire list.

Last, but not least: Don't forget to Tweet @Web2Market and use #AskMagentoExpert if you get stuck at any point. We're ready to help bring your sales up to speed.

10 QUICK DIY CONFIGURATIONS

1. Turn on Magento's built in caching.

a. Log into ADMIN
b. Go to SYSTEM ▶ CACHE MANAGEMENT
c. Select "ALL CACHE TYPES"
d. In the "ACTION" drop down menu, select "ENABLE"
e. Click "SUBMIT"

2. Enable "Merge JavaScript Files" and "Merge CSS Files" to reduce the number of http requests.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION
c. On the left hand side, go to "ADVANCED" and select "DEVELOPER"
d. Select "MERGE JAVASCRIPT FILES"
e. Change drop down to "YES"
f. Click "SAVE CONFIG"
g. Select "CSS SETTING"
h. Change "MERGE CSS FILES" to "YES"
i. Click "SAVE CONFIG"

3. Enable "Use Flat Catalog Category" and "Use Flat Catalog Product" to decrease database queries.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ CATALOG ▶ FRONTEND
c. Change "USE FLAT CATALOG CATEGORY" and "USE FLAT CATALOG PRODUCT" to "YES"
d. Click "SAVE CONFIG"

Forrester Research revealed that a majority of customers **expect a page to load in no more than 2 seconds**. If your Magento site takes longer, up to 50% of visitors will leave (and likely seek out a competitor).

4. Uninstall any extensions that you don't actually use.

a. Log into ADMIN
b. Go to SYSTEM ▶ MAGENTO CONNECT ▶ MAGENTO CONNECT MANAGER
c. Log in with your admin username and password
d. Select which extension you would like to install
e. In the "ACTION" drop down menu, select "UNINSTALL"
f. Click "SUBMIT CHANGES"

5. Limit the number of products on a product overview page.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ CATALOG ▶ FRONTEND
c. Adjust the number of products shown on pages.
d. Click "SAVE CONFIG"

47% of consumers expect a web page to load in 2 seconds or less.

10 QUICK DIY CONFIGURATIONS

📄 ▶ URL ▶ 🔗

6. Enable cron and log cleaning.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ ADVANCED ▶ SYSTEM ▶ LOG CLEANING
c. Change "ENABLE LOG CLEANING" to "YES"
d. Click "SAVE CONFIG"

7. Don't use category paths for product URLs.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ SEARCH ENGINE OPTIMIZATIONS
c. Change "USE CATEGORIES PATH FOR PRODUCT URLS" to "NO"
d. Click "SAVE CONFIG"

8. Use canonical link meta tag for products.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ SEARCH ENGINE OPTIMIZATIONS
c. Change "USE CANONICAL LINK META TAG FOR PRODUCTS" to "YES"
d. Click "SAVE CONFIG"

> *i* Aberdeen Research proved that a 1 second delay kills conversions by as much as 7%.

> *i* 40% of people abandon a website that takes more than 3 seconds to load.

🔗 ▶ URL

9. Use canonical link meta tag for categories.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ SEARCH ENGINE OPTIMIZATIONS
c. Change "USE CANONICAL LINK META TAG FOR CATEGORIES" to "YES"
d. Click "SAVE CONFIG"

10. Remove .html in the category URL suffix.

a. Log into ADMIN
b. Go to SYSTEM ▶ CONFIGURATION ▶ CATALOG ▶ SEARCH ENGINE OPTIMIZATIONS
c. Remove the ".HTML" and leave the area blank for "CATEGORY URL SUFFIX"
d. Click "SAVE CONFIG"

APPLICATION LEVEL FIXES

⚠ Ask a Magento Expert

The remaining tips will require some technical chops. If you feel confident, optimize away! Otherwise, ask your IT service provider or developer to help.

And you can always connect with us on Twitter @Web2Market and tag your question with #AskMagentoExpert.

11. Minimize redirects

IMPACT POTENTIAL: **HIGH**

WHY? Redirects trigger an additional HTTP request-response cycle and add round-trip-time latency.

12. Minify JavaScript

IMPACT POTENTIAL: **HIGH**

WHY? Compacting JavaScript code can save bandwidth and data transfer.

13. Leverage browser caching

IMPACT POTENTIAL: **HIGH**

WHY? Reduce the load times of pages by storing commonly used files from your website on your visitor's browser.

14. Enable gzip compression

IMPACT POTENTIAL: **HIGH**

WHY? Reduce the size of files sent from your server to increase browser transfer speed.

15. Minify CSS

IMPACT POTENTIAL: **HIGH**

WHY? Compacting CSS code can save many bytes of data and speed up downloading, parsing, and execution time.

i

Gomez and Equation Research found that as many as 75% of customers will turn to a competitor if a website is too sluggish.

16. Serve scaled images

IMPACT POTENTIAL: **HIGH**

WHY? Properly sizing images can save bandwidth and data transfer time on all devices.

17. Specify a "Vary: Accept-Encoding" header

IMPACT POTENTIAL: **HIGH**

WHY? Bugs in some public proxies may serve compressed versions of your resources to users that don't support compression. Specifying the "Vary: Accept-Encoding" header instructs the proxy to store both a compressed and uncompressed version of the resource.

18. Minify HTML

IMPACT POTENTIAL: **HIGH**

WHY? Compacting HTML code, including any inline JavaScript and CSS it contains, can save bandwidth and speed up processing.

APPLICATION LEVEL FIXES

□

19. Specify image dimensions

IMPACT POTENTIAL: **HIGH**

WHY? Specifying a width and height for all images eliminates unnecessary reflows and leads to faster rendering.

□

20. Avoid CSS @import

IMPACT POTENTIAL: **MEDIUM**

WHY? Using CSS @import in an external stylesheet can add additional delays to site loading times.

Mobile devices account for 20% of all e-commerce traffic, and its share is steadily growing.

□

21. Serve resources from a consistent URL

IMPACT POTENTIAL: **HIGH**

WHY? It's important to serve a resource from a unique URL to eliminate duplicate download bytes and additional round-trip delays.

□

22. Specify a cache validator

IMPACT POTENTIAL: **HIGH**

WHY? All static resources should have either a Last-Modified or ETag header. This allows browsers to take advantage of the full benefits of caching.

□

23. Avoid bad requests

IMPACT POTENTIAL: **HIGH**

WHY? Removing "broken links," or requests that result in 404/410 errors, gets rid of wasteful requests.

□

24. Avoid landing page redirects

IMPACT POTENTIAL: **HIGH**

WHY? Redirects on landing pages not only slow page loading speed, but while the redirections are occurring, the visitor sees nothing. In many cases, you can eliminate redirects without changing the function of a page.

□

25. Optimize images

IMPACT POTENTIAL: **HIGH**

WHY? Reduce load times by using appropriately sized images. For example, use JPEG or PNG images instead of the larger BMP or GIF filetypes.

□

26. Remove query strings from static resources

IMPACT POTENTIAL: **HIGH**

WHY? Most proxies, most notably Squid (up through version 3.0), do not cache resources with a "?" in their URL, even if a “Cache-control: public” header is present in the response. To enable proxy caching for these resources, remove query strings from references to static resources, and encode the parameters into the file names themselves.

□

27. Enable Keep-Alive

IMPACT POTENTIAL: **HIGH**

WHY? Enabling HTTP Keep-Alive or HTTP persistent connections allows the same TCP connection to send and receive multiple HTTP requests, thus reducing the latency for subsequent requests.

□

28. Avoid a character set in the meta tag

IMPACT POTENTIAL: **HIGH**

WHY? Specifying a character set in a meta tag disables the lookahead downloader in IE8. To improve resource download parallelization, move the character set to the “HTTP Content-Type” response header.

APPLICATION LEVEL FIXES

29. Defer parsing of JavaScript

IMPACT POTENTIAL: **HIGH**

WHY? In order to load a page, the browser must parse the contents of all <script> tags, which adds additional time for the page to load. By minimizing the amount of JavaScript needed to render the page, and deferring parsing of unneeded JavaScript until it needs to be executed, you can reduce the initial load time of your page.

30. Inline small CSS

IMPACT POTENTIAL: **HIGH**

WHY? Inlining small external CSS files can save the processing overhead of fetching these small files. Alternatively, combine several smaller external CSS files into one file.

31. Inline small JavaScript

IMPACT POTENTIAL: **HIGH**

WHY? Inlining small external JavaScript files lets you skip the overhead of fetching these small files. Alternatively, combine several smaller external JavaScript files into one file.

32. Minimize request size

IMPACT POTENTIAL: **HIGH**

WHY? Keeping cookies and request headers as small as possible ensures that an HTTP request can fit into a single packet.

33. Optimize the order of styles and scripts

IMPACT POTENTIAL: **HIGH**

WHY? Correctly ordering external stylesheets and external and inline scripts enables better parallelization of downloads and speeds up browser rendering time.

34. Put CSS in the document head

IMPACT POTENTIAL: **HIGH**

WHY? Moving inline style blocks and <link> elements from the document body to the document head improves rendering performance.

35. Specify a character set early

IMPACT POTENTIAL: **HIGH**

WHY? Setting the character set at the server level reduces browser processing time.

36. Combine images using CSS sprites

IMPACT POTENTIAL: **MEDIUM**

WHY? Combining images into as few files as possible using CSS sprites decreases the number of round-trips and delays in downloading other resources, reduces request overhead, also reducing the bandwidth usage of pages.

37. Prefer asynchronous resources

IMPACT POTENTIAL: **MEDIUM**

WHY? Fetching resources asynchronously prevents those resources from blocking the page load.

38. Add Expires headers

IMPACT POTENTIAL: **HIGH**

WHY? Expires headers let the browser know when it can grab a cached version of your web page. This helps you cut back on HTTP requests from the server.

39. Make fewer HTTP requests

IMPACT POTENTIAL: **HIGH**

WHY? Reduce HTTP requests through combined files and CSS sprites.

APPLICATION LEVEL FIXES

□ 40. Compress components with gzip

IMPACT POTENTIAL: **HIGH**

WHY? Save time and bandwidth by compressing resources, especially helpful for users with slower connections.

□ 41. Use a Content Delivery Network (CDN)

IMPACT POTENTIAL: **MEDIUM**

WHY? CDNs connect you to multiple server nodes spread across the globe, so you can give an equally fast web experience to users worldwide. Response and content download time plummet with a CDN.

□ 42. Minify JavaScript and CSS

IMPACT POTENTIAL: **MEDIUM**

WHY? Stripping unnecessary characters from JavaScript and CSS will speed up download times.

□ 43. Avoid URL redirects

IMPACT POTENTIAL: **HIGH**

WHY? Avoid using URL redirects as they add additional delays to a page load. Use a direct path instead.

□ 44. Reduce DNS lookups

IMPACT POTENTIAL: **LOW**

WHY? DNS lookups delay the initial request to a host. Making requests to a large number of different hosts can hurt performance.

□ 45. Use cookie-free domains

IMPACT POTENTIAL: **LOW**

WHY? Serve static content from a different domain to avoid unnecessary cookie traffic.

□ 46. Configure entity tags (ETags)

IMPACT POTENTIAL: **LOW**

WHY? Running multiple servers with default ETag settings can prevent 304 (Not Modified) responses.

□ 47. Avoid empty src or href

IMPACT POTENTIAL: **HIGH**

WHY? When a browser encounters empty src or href attributes, it makes an unnecessary request to generate the page, and can even end up crippling your servers with a large amount of unexpected traffic.

□ 48. Make AJAX cacheable

IMPACT POTENTIAL: **MEDIUM**

WHY? AJAX requests take place behind the scenes, so your visitor can't tell the difference between an AJAX request being slow, and nothing happening at all. Optimizing AJAX can lead to much faster processing.

□ 49. Put CSS at the top

IMPACT POTENTIAL: **MEDIUM**

WHY? Some browsers block progressive rendering of stylesheets near the bottom of a document. Your visitor gets stuck with a blank white page. Keep style sheets in headers to ensure progressive rendering.

□ 50. Remove duplicate JavaScript and CSS

IMPACT POTENTIAL: **MEDIUM**

WHY? Duplicate JavaScript and CSS files hurt performance by creating unnecessary HTTP requests (IE only) and wasted JavaScript execution (IE and Firefox).

APPLICATION LEVEL FIXES

73% of mobile internet users say that they've encountered a website that was too slow to load.

□ 51. Put JavaScript at bottom

IMPACT POTENTIAL: **MEDIUM**

WHY? You should place your JavaScript at the bottom of the page, right before the closing </body> tag. This way your page can load first while JavaScript loads in the background, giving the visitor a faster load time experience.

□ 52. Avoid AlphaImageLoader filter

IMPACT POTENTIAL: **MEDIUM**

WHY? Internet Explorer's AlphaImageLoader blocks rendering and freezes the browser while an image is being downloaded.

□ 53. Avoid HTTP 404 (Not Found) error

IMPACT POTENTIAL: **MEDIUM**

WHY? Fix up any 404 errors to avoid your users fetching content that does not exist. A 404 error is a sure-fire way to lose a customer.

□ 54. Reduce the number of DOM elements

IMPACT POTENTIAL: **LOW**

WHY? A complex page means more bandwidth usage and slower DOM access in JavaScript.

□ 55. Do not scale images in HTML

IMPACT POTENTIAL: **LOW**

WHY? Scaling images in HTML, as opposed to resizing the actual image file, consumes much more bandwidth and processing time. Resize your source files instead.

□ 56. Use GET for AJAX requests

IMPACT POTENTIAL: **LOW**

WHY? POST requires that headers and data be sent in two steps. GET sends headers and the data together in one go. Use GET instead of POST to collate headers and data.

□ 57. Avoid CSS expressions

IMPACT POTENTIAL: **LOW**

WHY? Expressions can produce thousands of unnecessary CSS evaluations.

□ 58. Reduce cookie size

IMPACT POTENTIAL: **LOW**

WHY? Keeping cookies as small as possible ensures that an HTTP request can fit into a single packet.

□ 59. Make favicon small and cacheable

IMPACT POTENTIAL: **LOW**

WHY? A favicon is stored in the favicon.ico file in the server's root. The smaller it is, the more quickly it can be retrieved. And making it cacheable means you can cut down on requests to the server.

□ 60. Make JavaScript and CSS external

IMPACT POTENTIAL: **MEDIUM**

WHY? Utilize browser caching and speed up page processing time by breaking out large inline CSS and JavaScript into external files.

▶ **You just conquered a mountain of Application level fixes! Aren't you glad there are only 5 Configuration Level Fixes to tackle next?**

CONFIGURATION
LEVEL FIXES

□ **61. Clean up Magento Database Log**

IMPACT POTENTIAL: **MEDIUM**

WHY? Magento's database can become cluttered very quickly if it's not cleaned up regularly. Cleanup can be performed manually, or better yet, configured to happen automatically.

□ **62. Avoid layered navigation**

IMPACT POTENTIAL: **LOW**

WHY? Avoid layered navigation unless absolutely necessary. It bogs down your site by consuming a large amount of resources.

□ **63. Disable the Magento log**

IMPACT POTENTIAL: **MEDIUM**

WHY? Magento, by default, logs many different things. Over time, the database fills up and performance slows, especially when traffic increases. And if you're using an external analytics tool, most of this logging is redundant.

□ **64. Name products with unique names**

IMPACT POTENTIAL: **HIGH**

WHY? If product names are too similar, users might not find what they're looking for in search. Using the same names can also lead to many URL rewrites bloating the system with thousands of duplicate records.

□ **65. Use Turpentine extension**

IMPACT POTENTIAL: **HIGH**

WHY? Turpentine enables Magento to work with Varnish, a rapid-cache reverse proxy. Turpentine and Varnish act together as a load-balancer and modify Magento to significantly improve the cache hit rate, boosting performance and handling high traffic.

SERVER
LEVEL FIXES

▶ **Right on! Now that your configurations are optimized, let's get your Server in line.**

□ **66. Use a dedicated server**

IMPACT POTENTIAL: **HIGH**

WHY? A dedicated server set up properly for Magento sites is the best way to run the resource intensive platform.

□ **67. Use a high performance drive SSD**

IMPACT POTENTIAL: **MEDIUM**

WHY? A Solid-State Drive or SSD can greatly reduce disk read and write times and deliver massive performance gains versus traditional spinning-disk hard drives.

□ **68. Use at least 2GB of RAM and a dual core CPU**

IMPACT POTENTIAL: **HIGH**

WHY? Not having enough server capacity to service your site is an obvious bottleneck to speed. This is a minimum standard for better performance across the board.

□ **69. Use separate web and database servers**

IMPACT POTENTIAL: **MEDIUM**

WHY? Don't make one server do double-duty. This greatly reduces performance, especially as your web traffic grows. Separate your web and database servers into two powerful machines to prepare for peak traffic. Also a requirement for PCI Compliance.

□ **70. Use Varnish Cache**

IMPACT POTENTIAL: **MEDIUM**

WHY? Varnish cache greatly accelerates Magento HTML processing, delivering much faster page loads.

SERVER LEVEL FIXES

□ 71. Use Turpentine extension

IMPACT POTENTIAL: **HIGH**

WHY? This caching extension lets you work in concert with server level cache to accelerate site speed.

□ 72. Tune default MySQL configurations

IMPACT POTENTIAL: **MEDIUM** / **HIGH**

WHY? There's a Perl script called MySQLTuner that will review your MySQL installation and automatically suggest adjustments to optimize performance. Optimizing your MySQL configuration has been shown to improve performance by as much as 65%.

□ 73. Modify PHP and .htaccess for optimal use for Magento

IMPACT POTENTIAL: **MEDIUM**

WHY? It is crucial that your .htaccess file and all PHP be configured efficiently and correctly to work with Magento. Search engine performance and page rank greatly depend on these being correct.

□ 74. Set up Cron for log maintenance

IMPACT POTENTIAL: **MEDIUM**

WHY? Magento logs every visitor's database requests. Over time, that can add up to a lot of records. Save time and speed up performance by automating log cleaning.

□ 75. Disable unneeded PHP mods

IMPACT POTENTIAL: **LOW**

WHY? Good housekeeping. Having mods enabled that are redundant or unused will only take resources away from tasks that actually need them.

If an e-commerce site is making $100,000 per day, a 1 second page delay could potentially cost you $2.5 million in lost sales every year.

□ 76. Engage compression

IMPACT POTENTIAL: **MEDIUM**

WHY? Enabling gzip compression will reduce bandwidth to end-users with slow connections, such as EDGE/3G cellular devices. This way, your visitors with slower connections still enjoy a great user experience.

□ 77. Set up Site crawler to warm Cache

IMPACT POTENTIAL: **MEDIUM**

WHY? By pre-loading pages into cache, customers will see faster first-time page loads.

□ 78. Use Redis

IMPACT POTENTIAL: **MEDIUM**

WHY? Unlike other caching software, Redis increases the speed of the Magento checkout by caching session and security sensitive pages.

□ 79. Use the most recent PHP version

IMPACT POTENTIAL: **LOW** / **MEDIUM**

WHY? Newer versions of PHP have been fine-tuned over time to run more efficiently. An out-of-date version of PHP or Magento can reduce performance by over 50%.

CONGRATULATIONS!

GET READY TO SPEED AHEAD OF THE COMPETITION

Well done! You just turbo-charged your Magento store.

You can feel great knowing you've given your Magento e-commerce website every possible advantage. Let us know what the experience was like, and feel free to contact us with any questions as you continue to grow your Magento e-commerce store.

> i
>
> **79% of web shoppers who encounter a slow e-commerce site won't shop there again, and over 40% warn their friends not to shop there either.**

Web2Market

WE'RE CRAZY ABOUT MAGENTO

Web2Market Managed Magento Hosting Does it All

We've made it our mission to help speed up the Magento eCommerce shopping experience, one Magento site at a time.

Web2Market's Managed Magento Hosting is your one-stop shop for Magento Hosting, Development, Extensions and Support.

Whether you're a startup web store just getting started or a savvy online brand processing thousands of orders a day, we can offer you a Magento hosting plan with the best value, speed, security, and support possible.

Contact us today to learn more about how Web2Market can power your Magento store and let's get crazy about Magento!

708-653-3100
info@web2market.com
web2market.com