



## 37 Point Magento Pre-Launch Checklist

Taking any website live can be a very stressful time as a developer. Your client is pushing to go live as soon as possible and the pressure can make it easy to forget steps. I've compiled some tips and tricks that I have picked up on [Meanbee's](#) projects for moving your Magento store from development to production.

- [Favicon](#)
- [HTML Head](#)
- [Placeholder Images](#)
- [Contact Information](#)
- [Transactional Emails](#)
- [Plain-text Passwords in Emails are Evil](#)
- [Change Admin Url](#)
- [Invoice and Packing Slip Design](#)
- [Tax Settings](#)
- [Shipping Origin](#)
- [404 Page](#)
- [Clear Test Data](#)
- [Product Inventory](#)
- [Flush JavaScript to 1 file](#)
- [Minify the CSS](#)
- [Flush CSS to 1 file](#)
- [Maintenance.flag](#)
- [Error Page Design](#)
- [Index Management](#)
- [Caching](#)
- [System Compilation](#)

[Google Analytics](#)  
[Google Base](#)  
[Google Sitemap](#)  
[Newsletter Subscriptions](#)  
[Shipping Methods](#)  
[Payment Gateways](#)  
[SSL Certificate](#)  
[Module Licenses](#)  
[PHP Configuration](#)  
[Turn on the Crons](#)  
[Log Cleaning](#)  
[Disable Unnecessary Modules](#)  
[Database Backup and Restoration Strategy](#)  
[“The Web should be fast”](#)  
[Be prepared](#)

## Base URLs

These are URLs that are stored in the database table core\_config\_data but can also be accessed through the admin interface if you have access to it: System → Configuration → Web. When you move the site, they will need to be updated otherwise all URLs will be referring back to the staging site. P.S. Don't forget the trailing forward slash!

## Favicon

Magento can be hard work at times, but that does not mean you can get away with making a fool out of yourself by forgetting to create a favicon.

## HTML Head

Do not leave the original settings in the HTML head. You should change the title and set the meta description and keywords, these are important for SEO. System → Configuration → General → Design → HTML Head.

## Placeholder Images

These are the images shown when a product does not have its own image. Create a default set for your site and place them in: catalog/products/placeholder/image.jpg, catalog/products/placeholder/small\_image.jpg and catalog/products/placeholder/thumbnail.jpg. Alternatively they can be uploaded in admin: System → Configuration → Catalog → Catalog → Product Image Placeholders.

## Contact Information

Ensure all the store information and email addresses are set. You will be able to find these in System → Configuration → General and System → Configuration → Store Email Addresses.

## Transactional Emails

Transactional emails use a different logo to the default one. Make sure that the image that you want to use is saved in your theme directory as logo\_email.gif.

Most of the emails will now be sorted due to using the configuration that you have already set. However, there are a couple of templates that as of version 1.4.1.1 still use hard coded values so it's necessary to go in to these and make some changes manually. To save you some time, the email templates that are affected are:

- Newsletter subscription confirmation – “Demo Store”
- New account confirmation key – “Demo Store”
- New password – “Demo Store”
- Share Wishlist – “Demo Store”
- New Order – “PST”

## Plain-text Passwords in Emails are Evil

It really gripes me when websites send out plain-text passwords in emails. Unfortunately, Magento does this as soon as you sign up for an account. So, I would personally recommend going into the transaction email for account confirmation and replace the password with some holding text e.g. “\*\*\*\*\*”. If people are interested in a “click-to-reset” password recovery module for Magento, [let me know](#).

## Change Admin Url

For security, it is best to change the URL that the admin interface resides on. This can be set in System → Configuration → Advanced → Admin → Admin Base URL

## Invoice and Packing Slip Design

Here, you will need to upload the logo that is to be used for invoices. Alternatively you can place logo\_email.gif in your images folder in the skin directory. There are a few other settings that you should pay attention to.

## Tax Settings

These should be all set up already, but it does not hurt to see if anything in the configuration was overlooked. These can be found in System → Configuration → Sales → Tax and also Sales → Tax.

## Shipping Origin

Directly related to the tax settings, the shipping origin needs to be set to the correct country in order to use the appropriate tax rates.

## 404 Page

Not to be pushed to the side of a development of any site. Remember to test it. Customers that find it will already be disappointed, do not let yourselves down by leaving the default Magento blocks still there. So, please remember to “Kill the dog”.

Also, in the default 404 page which can be found in CMS → Pages, there is reference to “Magento Store”. Ditch it.

## Clear Test Data

Remove all products, orders and customers that were created during development

## Product Inventory

When importing products the first time around, it is likely that you did not bother with managing stock. It's an unnecessary hassle when developing. Before you go live, check with your client what configuration he would like set up. You may have to do another product import with stock levels before going live.

## Flush JavaScript to 1 file

One essential measure to reduce loading time is by limiting the number of HTTP requests that are made each time. Magento stores can be heavy in JavaScript, but you can keep the requests down by flushing all of your JavaScript files into one: System → Configuration → Developer → JavaScript Settings.

You may have used the odd bit of inline JavaScript when developing. These will need to be moved in to their own file to avoid any unnecessary slowing of page load times.

## Minify the CSS

Use a [CSS compressor](#). Keep two versions of the css, one in readable form e.g. styles.css.orig and then have a minified version styles.css.

## Flush CSS to 1 file

This is the same idea as the JavaScript flushing, but be careful, it is currently an experimental

feature. Best to enable on the staging server before taking it live to see if it has any side effects. It can be found in System → Configuration → Developer → CSS Settings  
A useful extension available on [Magento Connect](#) is [Fooman Speedster](#). It combines, minifies and compresses all JavaScript and CSS.

## Maintenance.flag

Touch this file in the root of the Magento installation and it will display a holding page while you are going live. Especially useful if you are using something slow like FTP. Move this file across first and then queue the rest of Magento to transfer.

## Error Page Design

When there is an error on the site, or the maintenance.flag file is set, a different theme is used to display the error message. Without changing this your customers will see the Magento logo and default theme. To adapt this page to your own template you will need to modify the files located in the errors directory.

You will not want the error page to print a stack trace. To stop this happening open errors/local.xml (assuming you are working from the sample), and then change the action from “print” to “email”. You can then set an email address for it to send the error report to.

## Index Management

Make sure you have re-indexed after making any catalogue changes.

## Caching

Caching, although on by default would have been the first thing you disabled when beginning development. So, remember to re-enable and refresh it when you are ready to go.

## System Compilation

Another essential way to keep the loading time of your Magento site down is to compile it. Page loading times are said to have a performance increase of up to 50%, this is definitely not something you want to miss out on. You will be able to find settings for this in System → Tools → Compilation.

## Google Analytics

Your client is going to want to see how traffic fairs from day one. Magento does all the hard word for you (as per usual). Set up your [Google Analytics](#) account and then put your account number in System → Configuration → Sales → Google API → Google Analytics.

## Google Base

Another essential Google service that is already integrated into Magento. There are a few steps to do this, but [Inchoo](#) have written a nice [article](#) on it.

## Google Sitemap

No excuses for this one, my Grandmother could do it and she still thinks the postman delivers your emails (true story). Simply “Create Sitemap” in Catalog → Google Sitemap. Then, if you are using Google Webmaster Tools you can also point Google towards it.

There are further settings for this, e.g. to automatically regenerate the sitemap, and they can be found in System → Configuration → Catalog → Google Sitemap

## Newsletter Subscriptions

Test that customers can sign up and unsubscribe and that newsletters can be sent out. You don't want to be testing this process once you have real customers in the system.

## Shipping Methods

If you are using a fixed product or cart fee then make sure it is set. If you are going down the cart weight route then you should confirm that the logic is correct and that all products have weights attached to them

## Payment Gateways

Double check that your payment gateway is ready to jump from testing to live.

## SSL Certificate

This is one that is often “left on the shopping list”. Many payment gateways will not allow you to take the system live until you have one.

## Module Licenses

Some modules from Magento connect require you to enter a license key when setting up. Make sure that this key is still going to be valid when you move domains and obtain a new key if required.

## PHP Configuration

Make sure that you have reasonable values for `memory_limit` and `max_execution_time`. This

way, the heavier operations that Magento carries out can complete successfully. If either are too restrictive a product export/import will fail.

## Turn on the Crons

Some things in Magento require scripts to be run periodically in order to keep things up to date, e.g. Google Sitemap.xml. In order to configure these you will need to look in System → Configuration → Advanced → System and modify the Cron.php file in the root of your Magento installation. The Magento Wiki has a [page](#) on this.

## Log Cleaning

Magento keeps logs in the database, which, over time, can become bloated. There is an option in admin to clean the logs periodically, it can be found in: System → Configuration → Advanced → System → Log cleaning. This will need enabling as it is disabled by default, I recommend daily cleaning. Some articles suggest that use should optimise the log tables regularly as well in order to keep their size down.

## Disable Unnecessary Modules

It's not likely that your store uses every feature built in to Magento, so why have it enabled? You can disable modules in System → Configuration → Advanced. For example, ones that you might not be using include Mage\_Wishlist, Mage\_Newsletter, Mage\_Poll, Mage\_Rating and Mage\_Tag.

## Database Backup and Restoration Strategy

I'm not going to suggest any particular strategy, but take database backups *regularly*, and know how to restore when the time comes. There's even an [Automatic Database Backup](#) module to do it for you.

## “The Web should be fast”

Test your site out for speed. There are many different tools that you can use for this. I'd recommend either the [YSlow](#) extension for Firebug in Firefox, or [GTMetrix](#). These will then show your weak points and offer suggestions on how to make improvements.

Speed should still be a priority even in the broadband age. Remember that even with increasing internet speeds, there are still many on very limited connections. What's more, you are always being compared to other sites, so if you are too slow, your customers may move to a competitor.

## Be prepared

Once you have gone live, don't expect that to be the end of development. There are likely to be many bug fixes, editions and further development work. When developing sites with [Meanbee](#), we use a git repository and keep this up to date with all changes. This way we can always have two identical versions of the site; one that is public facing and the staging area. Once we are happy with changes we can push them live. With ssh access this could not be simpler with a *git pull* but becomes slightly more difficult if only FTP access is available. Unfortunately, this is often the case with managed hosting. A service that I was recently introduced to by [Nick](#) and we now use at Meanbee is [Deploy](#). Although still in Beta, it allows you to release the changes to the live site effortlessly even when using FTP. Marvellous.

Have you learnt from experience and want to add an item to this list? Share it, your Magento buddies will be most grateful.