

# Settings

Admin Page: **Design > Settings**

Note: Admin Page added in **Version 7.3.0**

The settings on this page control different aspects of how the design (templates, images, CSS, JS, etc.) work. Care should be taken when changing these, if something is done incorrectly it can result in a non-working website. Be sure to read the explanations for each of the settings below and you understand those explanations before changing the settings.

## Media Location Settings

<tip c w>**Caution:** Changing the settings in this section incorrectly can result in a non-working website, make sure you fully understand what each setting does before changing these.

Most sites should leave these settings as they are.</tip>

### Alternate External Media Base URL

Most sites should **leave this blank!** This is an advanced option which allows usage of a Content Delivery Network, for companies that have a CDN. It can also be used to specify an alternative domain name that is used exclusively for media such as images, see below for more info.

**If you do use it:** This should be set to the "prefix" to the template set folder (normally **geo\_templates/**), and should **include trailing slash**<sup>1)</sup>.

#### Usage Examples:

Examples below use the "thumbs\_up.gif" image in "my\_templates" template set, and using standard "geo\_templates/" folder for the template sets folder.

Alternate External Media Base URL	Generated URL	Notes
blank <sup>2)</sup>	geo_templates/my_templates/external/images/thumbs_up.gif	Uses relative URL, this is default and recommended for most sites.

Alternate External Media Base URL	Generated URL	Notes
//example.com/	//example.com/geo_templates/my_templates/external/images/thumbs_up.gif	Uses example.com for the base URL. Note that using // will make it use HTTPS on pages that are using secure connection. Make sure your alternate URL has SSL certificate working if you use SSL on your main site!
/	/geo_templates/my_templates/external/images/thumbs_up.gif	Changes it so that all external media uses / at beginning, so that they are now relative to the base of the domain. This is currently the best solution if it bugs you that pages with SEO re-written URLs look broken in Google cached version of your site.
/listings/	/listings/geo_templates/my_templates/external/images/thumbs_up.gif	See notes on /, you would do something like this if your software was installed in a sub-folder "listings" instead of at the root of the domain.

**Using alternate domain for media:** You can use a dedicated domain, that is used exclusively for media links, see the 2nd example above for how to use it. The **advantage** of doing this, is that it can help to reduce latency caused by un-used cookies sent with each request. More information can be

found from Google's own "best practices" at [Google Developer - Reduce Request overhead best practices](#).

**Using alternate SERVER for media:** Typically you would point the alternate domain to the same document root as the main domain and on the same server. But if you wish, we have made efforts to make the media files able to be loaded from a different server as well, as long as the server can make a DB connection to the same database as the main site. If you want to try this for your own site, look at the file **geo\_templates/min.php** and you will find PHP comments which will explain how to do this.

This is a very new ability and one that is not often used, so there may be snags, if you run into trouble using this feature please let us know!

## Template Set Folder

This is the folder used for template sets, normally this is set to **geo\_templates/**

This setting is changed by editing your **config.php** file, specifically the line:

```
define ('GEO_TEMPLATE_LOCAL_DIR', 'geo_templates/');
```

If you do not have that line in your config.php file, you may need to add it. Just change 'geo\_templates/' to the folder you would like to use, then re-name the folder using FTP. Note that re-naming this folder will require additional steps during the update process.

It should be set to the relative location of your geo\_templates directory, relative to the base installation location<sup>3)</sup>.

<tip c w>**Warning:** It is important that this is set properly in your config.php, or the website may not display properly. Make sure to back up your config.php file before editing it, so that you can restore the original file if anything goes wrong.</tip>

## JS Library Folder

This is the relative location of your js library directory, normally this should be set to **js/**

This setting is changed by editing your **config.php** file, specifically the line:

```
define ('GEO_JS_LIB_LOCAL_DIR', 'js/');
```

<tip c w>**Warning:** It is important that this is set properly in your config.php, or the website may not display properly. Make sure to back up your config.php file before editing it, so that you can restore the original file if anything goes wrong.</tip>

## Optimization Settings

These settings enable specific features that will help speed the page load. Note that some of the

features require certain things on the server, so not all sites will be able to use all optimization features. In most cases if the server is not capable, turning it on will simply not have an effect. But sometimes turning on a feature will cause 500 errors, in which case you will need to turn that setting off to work for your site.

<tip c n>**Note:** If turning on all of the settings causes the page to not format correctly, turn them all off, then turn them back on one at a time. After turning on each setting, check the site to make sure it still works. When it stops working, you will know which setting is not working on your server and can proceed accordingly.</tip>

## Combine, Minify, and Compress CSS and JS

This is the **main optimization** that will help greatly speed up overall page load times. It allows all CSS and JS to be combined so that it only loads a single "combined" CSS file, and a single "combined" JS file for the entire page, instead of loading a bunch of different files for each. With each additional "external file" that has to be downloaded, that can slow the page load down.

As it implies, the combined contents will also be "minified", meaning that extra contents are removed. It results in a single line download with all comments removed, meant solely for the browser to download as fast as possible.

<tip c n>**Note:** Keep this setting **turned off** until you have the design finished.</tip>

When you turn this on for the first time: View your website and browse the site! The "first time" someone looks at the site, it will have to generate the combined CSS and JS files, which can take a long time. So if you are the one that views the site first after this is turned on, it will load very slowly for you, but after that it should load lightning fast for you and everyone else. In other words, you do this so that the first "slow load" happens for you and not one of your normal site visitors.

## Troubleshooting

If turning this on breaks the site, the most common cause is file permissions. Use FTP to CHMOD the "geo\_templates/" folder to 755 and the "geo\_templates/min.php" to 644. This is necessary for most sites that use [suexec](#), such servers do NOT need to use CHMOD 777 to make a folder "writable", and doing so can actually cause server errors like this one.

## Clear Combined CSS & JS

This button will appear if the main combine setting is enabled.

Use this button whenever you make any CSS or JS changes on your design. Any time the CSS or JS changes, the combined copy will need to be re-generated. This is the reason we recommend to keep the combine setting turned off until your design is finished.

## Also Combine CSS and JS libraries

Only shows if [Combine, Minify, and Compress CSS and JS](#) is enabled.

We recommend to turn this on if you are already turning on the main combine setting, but the option is available if you have a reason to load libraries on their own (not combined).

## Use `{footer_html}` to delay loading of certain javascript

If this is enabled, JS will be loaded in `{footer_html}` rather than in `{head_html}`. If you use this, make sure that any of your custom JS is surrounded by `{add_footer_html}` and `{/add_footer_html}` to be moved to the end of the page, otherwise it may not work.

Note that this may not work with all custom JS, if you turn this on and it breaks your JS files, you may need to keep it off, or have a developer fix the custom JS to work when loaded at the end of the page.

**Why:** This will improve the "perceived page load time", so that it gives the appearance that the page loads faster. Most of the JS does not need to have the content loaded first to work, so by delaying that JS to load at the end, the main contents are loaded faster. The overall page load will remain the same since it will still need to load that JS, but it will appear to load faster.

## Troubleshooting

Turning this on will most commonly break custom javascript being used on the page. The first thing to try, is make sure you surround your custom JS with `{add_footer_html}` at the start, and `{/add_footer_html}` after the custom javascript. You may even notice those tags being used in the main software as well.

If using the `{add_footer_html}` does not fix it, or causes other problems, the custom JS may just need to be loaded inside of `jQuery()` call so that it is not run until the page is done loading. If you aren't sure how to do this, a developer familiar with 7.3 and higher should be able to help you with this.

If none of the above works, your custom JS may simply not work when it is delayed, and it may rely on other built-in scripts that are delayed... The result of this circular situation is that you cannot use the custom JS with this setting turned on. You can either remove the custom JS, or turn off the feature so that your custom JS will work. Ask yourself would you rather the page "seem to load" a little faster (especially on mobile devices), or would you rather your custom script worked? That is what it might boil down to as some JS may simply not be capable of being delayed.

## Use `.htaccess` for `geo_templates/`

This **Requires Apache**, if you are using IIS or other server this will most likely not work for you. Turning this on will allow you to enable the optimizations that will automatically generate an `.htaccess` file inside your `geo_templates/` folder for you.

## `.htaccess` - Stop Prying Eyes

Available when [Use .htaccess for geo\\_templates/](#) is enabled.

Turning this on will prevent anyone from looking at the directory index, meaning they will not be able to see what other files are inside of a specific folder. This will also prevent direct access to template files (ending in .tpl) directly from the web, for instance they won't be able to see the "un rendered" version of your home\_page.tpl file even if they know the direct URL to the file.

## **.htaccess - Compress Files**

Available when [Use .htaccess for geo\\_templates/](#) is enabled.

**Requires mod\_deflate.** If using Apache 2.4 (common on local WAMP like development installations), may also require **mod\_filter**.

This will add lines to the .htaccess that will use compression for most common "static" file types in geo\_templates. If it works on your server, we recommend turning this on.

### **Troubleshooting**

If Google Pagespeed or other tools report that you need to compress CSS or JS files, you may not have the Apache module "mod\_deflate" enabled and installed on your server. Contact your host or server administrator to have this installed.

If you are using a newer version of Apache, you may also need to have mod\_filter enabled as well.

If you get 500 errors, or if images / CSS / JS seem to not load, you may need to turn on mod\_filter. If that does not work or is not possible on your server, you may need to disable the compress files option.

## **.htaccess - Cache Files Longer**

Available when [Use .htaccess for geo\\_templates/](#) is enabled.

**Requires mod\_expires** Apache module.

If enabled, will add lines to the .htaccess file that will tell browsers to cache static CSS / JS files for about a month. This will help speed up page load times for repeat visitors to your site.

## **.htaccess - Use mod\_rewrite for Combined CSS/JS**

Available when BOTH [Use .htaccess for geo\\_templates/](#) AND [Combine, Minify, and Compress CSS and JS](#) are enabled.

**Requires mod\_rewrite** Apache module.

This one is a pretty cool feature, used in conjunction with the combined CSS/JS files. When this is turned on, it will use a URL like:

geo\_templates/.min/css/12.css

When you first visit the site, that file will actually *not exist yet!* It uses mod\_rewrite so that if the file does not exist yet, it actually uses

geo\_templates/min.php

It will use that file to generate the combined & minified CSS or JS file. What that will do is generate the combined / minified CSS or JS, then it will write the file to the .min folder. In other words, it will generate the contents and write the contents to the folder .min/css/12.css. Once it is generated, the next time someone views the page, since the file exists, it will load the static file, never even "touching" the script at all! So after the first time the combined CSS or JS file is loaded, after that it will be exactly the same as loading a static file, no overhead of loading a script.

## Advanced Settings

This is the advanced settings section.

### Do NOT Automatically Include default.css in Combined CSS Contents

Note: "Do NOT Automatically Include default.css in Combined CSS Contents" is the actual setting's name.

<tip c w>**Warning:** Turning this on will **break your site** unless you are following this tutorial<sup>4)</sup>. And as it mentions in the tutorial, using this option will make future software updates much more work.</tip>

This setting requires a little explanation and set up to explain why it is even in the software and how you can use it if you so choose, so bear with us here.

#### How System Loads CSS Normally:

TODO: Move this into the getting started section of the user manual somewhere.

**Normally**, the system loads the following CSS files automatically as **part of {head\_html} generated contents**

- normalize.css - Part of the normalize library
- default.css - From default template only(see below)
- theme1.css - only loaded if it exists
- theme2.css - only loaded if it exists
- custom.css - main CSS that you add your custom CSS to

Note that the **default.css** file is **always loaded from default template set**, no matter what. Even if you copy that file to your own template set, it will ignore it, the system is specifically coded to only load the **default.css** file from the default template set.

The **normalize.css** file is a 3rd party library, you can read more about it on the [normalize project page](#).

The **theme1.css** and **theme2.css** files can be used to apply 2 different theme (color) template sets.

The **default.css** will contain all CSS as-is as it comes with the software. You **do not edit** this file directly or copy it to your own template set to edit. Instead, you copy any sections that you wish to change, into your own **custom.css** file, and customize them there. Since the custom.css is loaded after the default.css, any CSS in custom.css will overwrite what is used in default.css. When you are doing so, remember that the less CSS you can have, the better, so be sure to take out parts that you aren't changing. The less CSS, the smaller the file, and the faster the page will load.

This system allows for optimum balance between reducing the size / amount of the CSS, making it easy to customize, and making it very update friendly. In the future, as new CSS is added for new features, you do not need to make any changes at all to your own custom.css file since any new CSS will automatically be part of the new default.css file! And unlike in versions before 7.3, you only copy in what is changed into your custom.css so it reduces the amount of "extra CSS" greatly.<sup>5)</sup>

### What the setting does when turned on:

It skips the step of loading default.css, so that only normalize.css, theme1.css, theme2.css, and custom.css files are loaded if they exist. So if you only have a custom.css, the only CSS to be loaded would be normalize.css and custom.css.

The typical use for this would be if your changes to the CSS file are so extensive, that you would be over-writing the majority of the CSS. In other words, if your own custom.css size is starting to approach the size of the default.css file, you may wish to make use of this if you don't mind the extra work required during updates.

Since default.css is not loaded, custom.css must have all CSS required! If you plan to use this option, you would start off by making a full copy of the default.css contents into your own custom.css file so that you have everything needed.

### Extra Steps during Update:

If this setting is turned on, you will have a lot more work on your hands for software updates. Every time you update, you will need to look at the changelog, where it lists every template change made in the new version. All changes listed for the default.css file, you will need to apply the changes to your own CSS. Failure to do this will result in new features looking broken, since you will be missing changes and new CSS required for the new features to function properly.

## Use Google Libraries API

This setting **Added in 6.0.0**, and moved to **Design > Settings**<sup>6)</sup> in version **7.3.0**.

Using this feature allows certain JS libraries to be loaded from the Google Libraries API instead of loading them from your local site. This provides 2 main advantages:

- Google Libraries API uses compression by default, if the browser is capable of handling it, so that the file downloaded is much smaller than if no compression was used. This makes the page load that much faster.
- Since it loads off of a common location (from Google's site), if the user has ever visited any site that uses the same library, they will already have the files in the browser cache, in which case it will not need to download anything!



Note that this only uses Google Libraries API for the *client side*<sup>7)</sup>, the admin will always load the JS libraries "locally". This is done in case loading it using google libraries is not working, so that the admin panel will still function to allow you to turn this feature off if needed.

[More Technical Info Here](#)

## CHMOD 777 Files

Setting moved to this page<sup>8)</sup> in version **7.3.0**.

Leave this box **checked** if you are not sure if you need it. If you know that your server / host uses what is known as [setuid](#), where scripts are "run as" your own user account, you will probably need to un-check the box.

If this setting is turned on, any file operations made to files and folders within `geo_templates/` folder, will be CHMOD to 777. This is necessary on some servers in order to be able to edit and over-write the file or folder using FTP, if you wish to edit the design by editing the files directly instead of through the admin panel, or if you ever wish to delete the files using FTP.

If this is not turned on, the file permissions will be set at whatever the default is for your server's configuration. On some servers, this may result in not being able to view, edit, or delete the file using FTP since the "apache user" is usually different than the "FTP user".

Note that this setting takes effect even if in **Standard Mode**, and it defaults to be checked (turned on).

## Allow Edit of system, module, and addon Templates

Setting moved to this page<sup>9)</sup> in version **7.3.0**.

If this setting is checked (turned on), in the admin panel, it will allow editing of system, module, and addon templates. Normally such editing is not allowed in the admin panel, since editing such files is not usually needed, and can make updates much more difficult.

Note that this setting **ONLY** has an effect when in **Advanced Mode**, and it defaults to be un-checked (turned off).

<tip c w>**Warning:** For your convenience, system, module, and addon templates can be edited in the admin panel when in advanced mode, and the setting is enabled in the advanced settings, but there are risks in customizing such templates, as noted below.

### Risks of Customizing System, Module, or Addon Templates:

- May require compatibility changes when updating the software. Such changes are the responsibility of the site owner and are not covered by the software update service provided by Geodesic Solutions.
- Problems caused by customizations are not covered by Geo support, if support finds that something works with the "default" templates but not with customized templates, they will not be able to help further.

</tip>

1)

Unless not using this setting, in which case set it to blank.

2)

Not actually set to "blank", but have the field blank as in the setting is empty

3)

Relative to the GEO\_BASE\_DIR setting, which is normally set automatically for you, to the base folder for the software.

4)

TODO: Create said tutorial and link to it here

5)

Older versions loaded theme\_styles.css, which was a full copy of the CSS, and also loaded theme\_styles.css from the default. That resulted in, worst case, double the CSS footprint with most if not all of the CSS being an exact copy. This resulted in double the CSS size of what it "had to be".

6) 8) 9)

previously the setting was found in **Design > Template Sets**

7)

what you see when you click **My Site** from the admin panel

From:

<https://geodesicsolutions.org/wiki/> - **Geodesic Solutions Community Wiki**

Permanent link:

[https://geodesicsolutions.org/wiki/admin\\_menu/design/settings/start?rev=1380557398](https://geodesicsolutions.org/wiki/admin_menu/design/settings/start?rev=1380557398)

Last update: **2014/09/25 16:55**

