How To Get Started With mod_pagespeed with Apache on a CentOS and Fedora Cloud Server

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: *http://faq.asphosthelpdesk.com/article.php?id=272*

One of the more recently popular modules for Apache is mod_pagespeed. It is an output filter for Apache 2.2+ that can be configured through a variety of options through configuration files or a .htaccess file. An "output filterâ€• is a something that transforms the data before it"s sent to the client. In other words, it"s a layer between your website and what the user"s browser receives when they visit your URL.

Speed Up the Web

The goal of mod_pagespeed is to speed up your website. It does this by applying filters to a variety of files in order to reduce the number of trips the browser has to make to grab what it needs, to reduce the size of those files and to optimize the length those files are cached.

Installation

Installation is very simple. It"ll vary depending on the operating system you use. Ubuntu and Debian have packages you can download and install (or any Linux distribution that uses .DEB packages). Other Linux distributions can download the source and build from that.

If you"re on a 64-bit version (likely)...

```
wget

https://dl-ssl.google.com/dl/linux/direct/mod-pagespeed-stable_current_x86_64.rpm
```

If you"re on a 32-bit version (less likely)...

```
wget
https://dl-ssl.google.com/dl/linux/direct/mod-pagespeed-stable_current_i386.rpm
```

Follow up with:

```
yum install at
```

(If you do not already have 'at' installed)

```
rpm -U mod-pagespeed-*.rpm
```

Remove the downloaded package

```
rm mod-pagespeed-*.deb
```

Note: Installing from source is outside the scope of this article. You can find detailed instructions from Google here: Build Mod_Pagespeed from Source

The module enables itself automatically when installed. However, you must restart Apache for it to start working.

```
/etc/init.d/httpd restart
```

You should now have a working version of mod_pagespeed up and running on your server. You can check this by looking at your page"s response headers. There should be a value for "X-Mod-Pagespeedâ€• with the version number you installed.

Setup

The installation package handles a lot of configuration out-of-the-box. In fact, there are conservative defaults that are automatically enabled on Apache. Depending on the Apache version you"re running, you"ll get a different version of the module installed and enabled. If you"re running Apache 2.2,`mod_pagespeed.so`will be installed; Apache 2.4 users will use`mod_pagespeed_ap24.so`.

Note: mod_pagespeed only works with Apache 2.2 and greater. There is also a bug with Apache 2.4.1 that prevents it from working with that version. Apache 2.4.2 or greater should be used.

Additionally, configuration files have been added to your Apache installation. The primary configuration file is `pagespeed.conf`.

This file is located at:

```
/etc/httpd/conf.d/
```

Configuration

If you wanted to, you could stop now. The defaults for mod_pagespeed are good, but you"ll often find that you can get better performance with a few additional tweaks to your site. Every site will get different results with different settings and it"s best to play around and find the settings that work best for you and your site.

For the purposes of this tutorial, we"ll go over a few of the more common settings.

How to configure mod_pagespeed

There are a few different ways mod_pagespeed can be configured. You can use the pagespeed.conf file described above to configure it for the whole cloud server. Or, if you"d rather, you can put your configuration settings in the VirtualHost directive for an Apache virtual host/website. Finally, you have the option of specifying directives in a .htaccess file, such as what most sites do for mod_rewrite.

The least performant of these options is the .htaccess file because it has to be loaded with every request. The pagespeed.conf file is loaded when Apache starts, so it"s the ideal place to store your configuration settings. Inside the VirtualHost directive is also preferable to inside your site"s htaccess file for the same reason. That"s a good place to put site-specific settings too.

You can use whatever text editor you want to edit the configuration file. For this tutorial, we"ll be using nano.

To start editing the main configuration file, use the following command:

```
nano /etc/httpd/conf.d/pagespeed.conf
```

Basic Settings

In general, the settings in pagespeed.conf are pretty well documented inside the file. There is also a great list of filter examples available from http://www.modpagespeed.com. Here are a few common settings you might want to play with to optimize for your site"s performance.

Turn mod_pagespeed On/Off

First off, you can turn the module on or off with the ModPagespeed setting.

```
ModPagespeed on
```

or

```
ModPagespeed off
```

Rewrite Levels

You can specify different "levelsâ€• of settings to simplify any configuration. The default is `"CoreFilters.â€•`It contains a set of filters the Google team believes is safe for use. The filters are the individual actions that are applied to a file. In general, you won"t need to change this value. It"s easier to use this default and then enable or disable filters using the`ModPagespeedEnableFilters`and `ModPagespeedDisableFilters`directives.

The default setting:

```
ModPagespeedRewriteLevel CoreFilters
```

To disable CoreFilters use this setting:

```
ModPagespeedRewriteLevel PassThrough
```

Note: You"ll have to explicitly enable any filters you want to turn on using the "`PassThrough` â€• setting.

Using the default "CoreFiltersâ€• rewrite level includes a number of filters by default. As of the time of this

writing, it includes:?

```
add_head
combine_css
convert_jpeg_to_progressive
convert_meta_tags
extend_cache
flatten_css_imports
inline_css
inline_import_to_link
inline_javascript
rewrite_css
rewrite_images
rewrite_javascript
rewrite_style_attributes_with_url
```

New filters will be added in the future. By using CoreFilters, you"ll automatically have these filters enabled if they become part of the default set whenever you update mod_pagespeed. Using PassThrough will require you to explicitly enable the new filters.

Enable Filters

If you"d like to enable additional filters, you can pass them as a comma-separated list to `ModPagespeedEnableFilters`. You can have multiple ModPagespeedEnableFilters directives throughout your configuration files. So, if you want to enable a filter per site, you could enable it in the virtual host configuration file or in the .htaccess file instead of in the main pagespeed.conf file.

Here"s an example that enables the Pedantic filter (which adds the type attribute to script and style tags) and the Remove Comment filter (which removes all HTML comments):

```
ModPagespeedEnableFilters pedantic,remove_comments
```

Disable Filters You can also disable filters on a per-case basis if you"d like. Specify a list of filters you"d like to disable similar to

```
ModPagespeedEnableFilters
```

The following example disables the "Convert JPEG to Progressiveâ€• filter even though it"s part of the CoreFilters set:

```
ModPagespeedDisableFilters convert_jpeg_to_progressive
```

Specify Which URLs are Rewritten

By default, mod_pagespeed rewrites everything it can. You can disable certain files (for example Javascript libraries) from being rewritten with the following directive:

```
ModPagespeedDisallow "*/jquery-ui-*.min.js"
```

This would disable rewriting of any files that match the wildcard pattern specified (jquery UI in this case).

You can also turn off the rewriting of all files by default and only enable files you want to rewrite manually. You can do this with the following settings:

```
ModPagespeedDisallow "*"
ModPagespeedAllow "http://*digitalocean.com/*/styles/*.css"
ModPagespeedAllow "http://* ortal
```

.com/*.html"

```
/etc/init.d/httpd restart
```

The order of execution means that all files at asphostportal.com ending in .html would be rewritten. That last Disallow directive means any URLs matching that pattern would not be rewritten because it overrides the previous setting.

Restart Apache

Don"t forget if you"re using the pagespeed.conf or VirtualHost files to alter the settings, you"ll have to restart Apache for the settings to take effect. You can do this with the following commands:

```
/etc/init.d/httpd restart
```

Conclusion

This guide will help you get started using mod_pagespeed. There are a number of other settings and directives that can be applied server-wide or per-site. In addition, mod_pagespeed is under active development so it"s changing every day. For more detailed information, visit the Google-run http://www.modpagespeed.com.