



Healing Thresholds

Connecting Community and Science to Heal Autism



[Autism therapies](#)

[Find service provider](#)

[What's new](#)

[About us](#)

Search

[Home](#) > [Administer](#) > [Site configuration](#) > [Performance](#)

Performance

[Performance](#)

[Boost Settings](#)

[Boost htaccess rules generation](#)

Boost File Cache

Boost - Static page cache:

Disabled

Enabled

(Not Recommended) Set Boost & Core (if enabled) cache for each page

Static page caching is a mechanism that stores dynamically generated web pages as HTML files in a special cache directory located under the Drupal installation directory. By caching a web page in this manner, the web server can serve it out in the fastest possible manner, without invoking PHP or Drupal at all. While this does provide a significant performance and scalability boost, you should note that it could have negative usability side-effects unless your site is targeted at an audience consisting mostly of "anonymous" visitors.

Gzip page compression (Boost & Core):

Disabled

Enabled

By default, Drupal compresses the pages it caches in order to save bandwidth and improve download times. This option should be disabled when using a webserver that performs compression.

Boost - HTML - Default maximum cache lifetime:

12 hours

The maximum cache lifetime is the maximum amount of time that will elapse before the cache is emptied. Cache lifetime gets checked on cron runs. Flushing the content before it is expired can happen like for example when a node is edited.

Boost - XML - Default maximum cache lifetime:

1 hour

The maximum cache lifetime is the maximum amount of time that will elapse before the cache is emptied. Cache lifetime gets checked on cron runs. Flushing the content before it is expired can happen like for example when a node is edited.

Boost - JSON - Default maximum cache lifetime:

1 hour

Enable the caching of this content type to enable this selection box.

0 Clear ALL Boost cached data: 14208 pages

1 Clear Boost expired data: 8566 pages

Boost cacheability settings

Cache pages that contain URL Variables

Boost will cache pages that end with "?page=1" among others (anything with a "?" in the url).

Cache html documents/pages

Boost will cache most Drupal pages.

Cache .xml & /feed

Boost will cache .xml and /feed urls as xml data.

Cache ajax/json

Boost will cache ajax/json responses.

Cache .css

Boost will cache CSS files.

Cache .js

Boost will cache javascript files.

Statically cache specific pages:

Cache every page except the listed pages.

- Cache only the listed pages.
- Cache pages for which the following PHP code returns TRUE (PHP-mode, experts only).

Pages:

autheal/*

Enter one page per line as Drupal paths. The "*" character is a wild-card. Example paths are 'blog' for the blog page and 'blog/*' for every personal blog. '<front>' is the front page. If the PHP-mode is chosen, enter PHP code between '<?php ?>'. Note that executing incorrect PHP-code can severely break your Drupal site.

Boost directories and file extensions

Cache Dir: *

cache

Do not store the cache file path in the database

Enabling will allow for correct multi-site caching, in cases where different content is served from the same Drupal installation, based on domain. Examples: Multi-site with a single/shared database, site translation detection based on domain, and the contributed "Domain Access" module.

Generated output storage (HTML, XML, AJAX)

Static storage (CSS, JS)

Boost advanced settings

Clear expired pages on cron runs:

- Disabled
- Enabled

Caution - If disabled you will need some other method of clearing cached pages. If Disabled, files have to be manually deleted from server or cleared using the administrative interface ([Enable the Boost: Pages cache status](#) block).

Check database timestamps for any site changes. Only if there has been a change will boost flush the expired content on cron.

Pre-process function:

The name of a PHP function used to pre-process the contents of each page before writing them out to static files. The function is called with the contents of the page passed as a string argument, and its return value is used as the data written out to the disk.

Only allow ASCII characters in path

Only allowing ASCII characters is a safe way to cache pages. It severely limits i18n support so this can be turned off. Fair warning, disabling this may cause "page not found" errors depending on your url structure (spaces are bad, ect...). If you follow RFC 3986 you should be ok.

Flush all sites caches in this database (single db, multisite).

This will flush/expire all cached files stored in this database, instead of only being specific to this site. Useful for i18n sites. You need to setup a separate cron call for each database (in your multisite install) either way though. This covers shared database usage; or place the boost tables into the a shared database, to have this setting work in a multiple database environment.

Asynchronous Operation: output HTML, close connection, then store static file.

Run php in the background. When a cached page is generated, this will allow for faster page generation; downside is the headers are not the standard ones outputted by Drupal; sends "Connection: close" instead of "Connection: Keep-Alive".

Clear all empty folders from cache.

Disable this if you have to set settings for each dir/subdir, due to the way your server operates (permissions, etc...).

Clear all cached pages referenced via CCK with a node on insert/update/delete

The [Node Referrer](#) module is recommended.

Clear all cached terms pages associated with a node on insert/update/delete

Works with view's taxonomy/term/% path as well as core.

Clear all cached pages in a menu on an insert/update/delete operation:

- Disabled
- Only Flush Menu Parents, Siblings & Children
- Flushes Entire Menu Tree

This can flush a lot of pages depending on your menu structure.

Clear all cached views pages associated with a node on update/delete

Clear all cached views pages associated with a node on insert

WARNING: This can be very slow, all views get run to find out where this node lives.

Clear Boosts cache when site goes offline

Under site maintenance when the status is set to offline, boost clears its cache. If you do not want this to happen, clear this checkbox. Pages that are not cached will still send out a Site off-line message, so be smart if turning this off.

Overwrite the cached file if it already exists

This is useful if crawling a site before it goes live.

Do not cache if php error on page

Selected - Do not cache the page if there are PHP errors. Not Selected - Cache pages even if it might contain errors.

Do not cache if a message is on the page

Selected - Do not cache the page if there are Drupal messages. Not Selected - Cache pages even if it might contain a Drupal message.

Turn off clean url's for logged in users

Drupal will output non clean url's for non anonymous users. This allows for the browser to cache the page and still have logging in work. This is more on the extreme side of tweaks.

Aggressive Gzip: Deliver gzipped content independent of the request header.

In order to deliver gzipped content independent of the header, this will test for gzip compression in a small iframe by sending it compressed content. This compressed content is javascript which creates a cookie with a note of gzip support. On the server side it checks for the cookie and then sends out gzipped content accordingly. See [Website Performance - Activate Gzip](#) . In short some firewalls/proxies mangle the gzip header; this gets around that. iframe is on non compressed version of the frontpage only.

Files: Enter in a 4 digit number (octal) that will be used by `chmod()` . Example 0664:

Sometimes because of funky servers you need it use a different file mode then the default.

Directories: Enter in a 4 digit number (octal) that will be used by `chmod()` . Example 0775:

Sometimes because of funky servers you need it use a different file mode then the default.

Expire content in DB, do not flush file.

Instead of flushing the file ASAP this will set the database entry to expired. File will be flushed on next cron run. This is more on the extreme side of tweaks.

Ignore cache flushing:

- Disabled
- Only Ignore Clear Entire Cache Commands (Recommended if caching css/js files)
- Ignore Clear Entire Cache Commands & Cron Expiration
- Ignore All Delete Commands (Not Recommended)

Make a selection to put your site into a static cached state. Recommend turning on CSS & JS caching if enabled.

Watchdog Verbose Setting:

Disable warning about reaching the ext3 file system subdir limit.

Disable the warning on the status page about coming close to the file system directory limit; warning thrown when at 31,000 subdirectories. Boost will stop creating new subdirectories when 31,998 subdirectories have been created. Can be a slow operation & might even timeout; if your status page will not load, Disable this warning.

Boost Retro Mode (no database)

Boost crawler

Enable the cron crawler

Pre-cache boosted URL's so they get cached before anyone accesses them. Enable the crawler first and save settings to use Preemptive Cache settings.

Do not flush expired content on cron run, instead recrawl and overwrite it.

Bypass cron cache expiration and load to be expired content in the crawler. Allow this servers IP to bypass the boost cache. Check for fast but potentially stale; unchecked for slow but always fresh.

Preemptive Cache HTML

Crawl Site after cron runs, so the cache is primed.

Preemptive Cache XML

Crawl Site after cron runs, so the cache is primed.

Preemptive Cache AJAX/JSON

Enable the caching of this content type to enable this checkbox

Crawl All URL's in the url_alias table

Preemptively cache all urls found in the Drupal url_alias table. This will crawl that page even if it is not expired. Enable & run cron to get the boost cache loaded.

Number of URLs to grab at a time when loading up the crawler table:

You might need to adjust this (shrink) if you get mysql errors when starting the crawler. Increase if your trying load up the crawler with a lot of URLs, as this can make the operation be faster.

Crawler Throttle:

Wait X micro seconds in between hitting each url. 1000000 is 1 second.

Crawler Batch Size:

Number of URL's each thread grabs per database operation.

Number Of Threads:

Be careful when choosing more than 2 threads.

[Boost crawler - Live info](#)

Estimated Time to crawl site - highly inaccurate:

Boost Apache .htaccess settings generation

[Explanation of .htaccess variables](#)

Be sure to save the configuration and then go to the [htaccess rules generation page](#) and copy the rules.

Servers URL or Name:

- %{SERVER_NAME}
- %{HTTP_HOST}
- autism.healingthresholds.com

Best to leave these as %{}, only try the last option(s) if boost is still not working.

Document Root:

- %{DOCUMENT_ROOT}
- /var/www/autism.healingthresholds.com

Value of `%{DOCUMENT_ROOT}` is recommended for this server.

Etag Settings:

- Do Nothing
- Set FileETag 'None' - Do not send an etag
- Set FileETag 'All' - Default if enabled
- Set FileETag 'MTime Size' - Useful in server clusters (Recommended)

Uses [FileETag Directive](#) to set [ETags](#) for the files cached by Boost. [More info on this subject](#)

Boost Tags:

- Inject boost tags at the bottom of the file
- Set Header and tags
- Only Set the header

In order to identify that the page is being served from the cache Boost can: Set a comment at the bottom of the page, indicating that this page is cached by boost; It can also send out a header that will identify any files served from the boost cache; Or it can do both.

Clear Boost's Database & File Cache

Warning: This gives you a clean slate for the boost database & file system, use with caution. If you change the directories or file extensions (in "Boost directories and file extensions"), then pressing this button would be a good idea, but not required. If you changed the CSS or JavaScript and wish to push it out, you need to press this button.