

Performance

To access: Click the **Performance** icon  on the [Admin Panel](#) or Access <http://example.org/tiki-admin.php?page=performance>

By default, Tiki is configured with default performance settings which should suit most contexts. For example, Tiki uses the [Smarty template engine](#) and it has [built-in cache](#). If you find Tiki slow, this page is here to help.

Basic tips

- Don't be cheap on hardware or hosting. Get the good stuff.
- Use recent versions of PHP/MySQL/Apache etc. [as each new version is typically faster than the previous](#)
 - [PHP 5.3 is 1.2 times faster than PHP 5.2 for raw execution speed](#)
 - Use the latest stable Tiki version or LTS version (but you should be doing this anyway even without performance concerns)
- Use OPcache, which is now the de facto standard starting in PHP 5.5 (and installable on older versions). If that is not working, use APC or XCache (see below)
- Turn off any feature you are not using
- Use the [Performance Admin Panel](#) at tiki-admin.php?page=performance where you can configure the various performance-related settings, including [Content Delivery Network](#). Otherwise, most performance tuning is done by tweaking the server (outside the scope of Tiki)
- Increase the "memory_limit" of your PHP instance (128M is the default)
- Think also about configuring your web server (for example, setting a long expiry date for images, css and js files)
- [Tiki5](#) has many enhancements related to [YSlow](#) tests
 - Activate [htaccess](#)
- <http://blog.lphuberdeau.com/wordpress/2010/03/24/improving-rendering-speed/>
- Use a free tool like gtmetrix.com to measure the impact of your optimisations on the loading speed of your site.
- If you run into trouble, check [Troubleshooting](#).

Real User Measurement

- [Real User Measurement](#)

Minify JavaScript

- [Minify JavaScript](#)

Minify CSS

- [Minify CSS](#)

Apache (httpd.conf)

- [httpd.conf](#)

PHP Settings (php.ini)

- [php.ini](#)

Accelerating PHP

OPcache

- [OPcache](#)

Alternative PHP Cache (APC)

- [APC](#)

XCache

- [XCache](#)

eAccelerator

- [eAccelerator](#)

Memcached

- [Memcached](#)

Redis

- [Redis](#)

Database performance

This is a [troubleshooting](#) guide for database performance.

MySQL

MySQL has a slow queries feature which is accessible via a log or at the command line. Please [read the MySQL manual page](#) to enable this feature.

Using the information in MySQL's [chapter on optimization](#), look for places where indexes may improve the database access speed. Add an index, then re-test. If you think you've found an improved indexing scheme, please open a tracker item on [dev.tiki.org](#).

In `tiki-admin.php?page=general`, you can activate **Log SQL** and set **Log queries using more than (seconds):**

MySQL replication

- [MySQL replication](#)

Related pages

- [Compression](#)
- [Link Cache](#)

Related links

- <http://www.webpagetest.org/>
- <http://www.showslow.com/>
- <http://gtmetrix.com/>
- <https://www.dotcom-tools.com/>
- <https://www.ohloh.net/tags/stress-testing>
- [mod_perl Performance Tuning](#) (This is a page about mod_perl tuning, just ignore the perl stuff and look at the apache tweaks in there.)
- [Hauptling Schnelles Wiesel: Wirksames Tuning für viel besuchte Webauftritte](#) Additional apache tuning for sites with much traffic (in German)
- <http://www.joedog.org/siege-home/>
- <http://blog.lavoie.sl/2012/09/protect-webserver-against-dos-attacks.html>

Alias

- [Tikiwiki Performance Tuning](#)
- [Database performance](#)