

Server check

Tiki Check is stable & functional. Below are some ideas for the future, but no more imminent work is planned.

Bugs

- 9.2 Tiki check was reporting "You do not have the recommended PDO database driver/abstraction layer. You do have the MySQLi driver though, so we will fall back to the AdoDB abstraction layer that is bundled with Tiki.", yet an issue was reported:
http://irc.tiki.org/irclogger_log/tikiwiki?date=2012-12-19,Wed&sel=120#l116
- 10.x "Notice: Undefined index: max in /www/htdocs/w00ec615/10x/modules/mod-func-facebook.php on line 57" <http://demo.tiki.org/18x/tiki-check.php>

Next

- [tiki-check.php](#): make all these 50+ values available to Nagios/Icinga/Shinken
- tiki-check.php should refuse being indexed by robots (how to do this in PHP?)
- The checks we have for [htaccess](#) (which are in tiki-admin.php?page=sefurl) should be available in tiki-check.php and when .htaccess is not the correct one for that version, there should be a huge warning
- Make it easy to find for site admins (ex.: add link in menu, or admin panel)
- Backport to branches/9.x
- ~~tiki-install.php~~ -> * PHP error reporting level should not be reverted Done in trunk and 9.2
- [Testing Tiki installations on major Shared Hosting companies](#)
- ~~Make a section for advanced things that basic users shouldn't worry about~~ Done, I changed the category from bad to info

Questions

- In tiki-admin_system.php, when exporting a zip: Error:PCLZIP_ERR_READ_OPEN_FAIL (-2) : Unable to open archive "" in wb mode
- Is [realpath_cache](#) check suitable for Tiki Check?
- Is there a reliable way to tell people where to change the settings? (ex: /etc/php.ini /etc/my.cnf)
- Are warnings about default_charset and date.timezone too stern? I have seen them on servers where Tiki works nicely
 - I changed from bad to ugly
- if possible warn when running with another MPM than "prefork" - PHP is NOT thread-safe
 - can this be checked from PHP at all? (without using exec and parsing apachectl output)
- session.save_path = /tmp
 - Does it make sense to check that?! Reasonable values for this vary from server to server and setup to setup.
- Can we check for a working .htaccess?

- e.g. create a htaccess with some RewriteRule in it, call the URL and see if it is being rewritten
- any smarter way?
- Can we check for the configuration of mod_deflate/expires/security ?
- Disk space check should be KILLED completely, it's the wrong place for it here.
 - Is there a more reliable way to detect disk space available, which seems unreliable because of quotas and on shared hosting.
 - This is really something the [Monitoring](#) system should warn about 😞
 - <https://github.com/zendframework/ZendDiagnostics#diskfree>

Phase 2

- [Zend_Search_Lucene doesn't work on certain servers](#)
 - Also, sometimes, it crashes during rebuild. If you have two directories (temp/unified-index-new/ and temp/unified-index/, it means that something went wrong. Tiki Check should detect and report this.

Beyond Tiki Check, but let's explore

- Should we have tiki-admin.php?page=server where we attempt to override various settings uncovered by Tiki Check?
 - [Increase memory_limit to 128M if detected below that value](#)
 - Done in <http://sourceforge.net/p/tikiwiki/code/43870> and <http://sourceforge.net/p/tikiwiki/code/43907>
 - Should we turn off magic quotes if we can? `ini_set('magic_quotes_gpc', 'Off');`
- File storage
 - If more than 50 Megs in MySQL, recommend to the admin to move to filesystem
 - If set to filesystem, but they are not accessible/writable, give a warning
- check if we are often using more than x% of memory_limit and warn the admin
- Run tiki-check automated, regularly in the background and e-mail admin or something like that
 - Run on [Pre-Dogfood Servers](#) and send an email report of hard crashes. (email could be stored in db/local.php)
- security - how to lock it? (turtle)
 - If a Tiki is installed where tiki-check.php is, it should use tiki auth system and check tiki_p_admin, or ask for db/local.php info (because if Tiki is crashed, it can't check for tiki_p_admin). If it detects a brute force password attack, it should lock itself like t0iki-install.php does.
- what happens, if Tiki is crashed
 - We want it to run in standalone, but we don't want people to DoS a Tiki to read it
- A function to export to txt, csv, html or something for easy distribution to people wanting to help on debugging
 - ask LPH for some parsable format
 - Could be .csv to mother.tiki.org
 - We should do this at the end
- In tiki-admin.php it should be warned that activating a feature doesn't help, if the server environment

doesn't support it

- e.g. [fileinfo extension](#) is needed for 'Validate uploaded file content'

File/folder permissions

- We currently have a check somewhere (tiki-install.php?) which brings up an ugly page if perms are detected to be incorrect and informs the user on what to do next
- http://demo.tiki.org/trunk/tiki-admin_security.php?check_file_permissions
- <http://tikiwiki.svn.sourceforge.net/viewvc/tikiwiki/trunk/permissioncheck/>
- In some places, Tiki code checks if it can write somewhere. Ex.: tiki-edit_templates.php, tiki-edit_css.php
 - Maybe these checks should be centralized so we can have an overall diagnostics?
- several different usecases need different treatment in setup.sh
 - if Tiki install fails, we can use permissioncheck (at work, 2012-10) to guide the users

tiki-recovery.php

- See [Recovery](#)

Nice to have

- Can Tiki index file content? -> [Search Within Files](#)
 - Link to Admin->File Galleries
- Check if Tikis Paths fit [open_basedir](#)
- It would be nice to run from the command line php tiki-check.php and have a decent looking output (now it's hard to read HTML)
 - Since CLI PHP can be different than the web PHP, is this a good idea?

Low priority (only do if very easy)

- Show DB-credentials form in the MySQL section
- Is it possible to report 32bits vs 64 bits? (useful to know what software to install)
- Indicate if MyISAM or InnoDB (people can use adminer.php)
- Warn if correlating values are too far off each other, for example
 - max_execution_time <> max_input_time
 - upload_max_filesize <> post_max_size <> max_allowed_packet

sh tiki-check.sh

- Since some things (like [Apache worker MPM](#)) can't be detected via PHP, perhaps the next step will be tiki-check.sh Let's move all the checks that can't be done via PHP to this section. And even here, some things will require root.
- Check for these [Server tools](#) or for [requirements for TRIM](#)

Server check script examples

- <https://github.com/llaville/php-compat-info>
- <http://www.bitweaver.org/wiki/bitweaverRequirements>
- <http://www.contao.org/en/configuring-the-live-server.html>
- <http://www.magentocommerce.com/knowledge-base/entry/how-do-i-know-if-my-server-is-compatible-with-magento>
- [WordPress ServerBuddy](#) (needs a WordPress install)
- <http://phpsysinfo.sourceforge.net/>
 - demo: <http://phpsysinfo.sourceforge.net/phpsysinfo/index.php?disp=dynamic>
- <http://phpsec.org/projects/phpsecinfo/> (very old)
- <http://www.phpservermonitor.org/>
- <https://packagist.org/packages/linfo/linfo>

Good, bad, ugly references

- <http://en.wikipedia.org/wiki/GStreamer#Plug-ins>
- http://en.wikipedia.org/wiki/The_Good,_the_Bad_and_the_Ugly

In Nagios, it's [OK](#), [WARNING](#), [UNKNOWN](#), or [CRITICAL](#)

Related

- [Reporting errors \(to tiki.org\) system \(like Firefox Talkback\)](#)
- [Update notifier](#)
- [Infrastructure Revamp](#)
- [Cron](#)
- [Compatibility](#)
- <https://docs.zendframework.com/zend-diagnostics/diagnostics/>

alias

- [System check tool](#)
- [pre-flight check](#)
- [Test server compatibility](#)
- [Check](#)