

## Bugs & Wish list

Installer not ready for PHP 7.4 in Tiki 18.x | Tiki Wiki CMS Groupware :: Development

Installer not ready for PHP 7.4 in Tiki 18.x

### Status

✖ Closed

### Subject

Installer not ready for PHP 7.4 in Tiki 18.x

### Version

18.x

### Category

- Consistency

### Feature

Installer (profiles, upgrades and server-related issues)

### Resolution status

Invalid

### Submitted by

hman

### Lastmod by

hman

### Rating

★ ★ ★ ★ ★ (0) ?

### Related-to

- ✖ Migrating 18.8 to 21.4 fails utterly, breaks layout

### Description

I have migrated one of my Tikis from my personal webspace to one of an organization for which I admin the IT, and therefore I did have a look at the installer (but edited db/local.php instead).

The installer throws several warnings, that ~~might~~ will become errors in the next PHP versions. This is what you get on PHP 7.4:



```
Deprecated: Array and string offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/lib/init/initlib.php on line 221 Deprecated: Array and
string offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/vendor_bundled/vendor/h5p/h5p-core/h5p.classes.php
on line 2590 Deprecated: Array and string offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/vendor_bundled/vendor/h5p/h5p-core/h5p-
development.class.php on line 70 Deprecated: Array and string offset access syntax with curly
braces is deprecated in /homepages/0/MYHOME/htdocs/tiki-18.8/lib/tikilib.php on line 414
Deprecated: Array and string offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/lib/tikilib.php on line 1858 Deprecated: Array and string
offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/lib/tikilib.php on line 4483 Deprecated: Array and string
offset access syntax with curly braces is deprecated in
/homepages/0/MYHOME/htdocs/tiki-18.8/lib/tikilib.php on line 4483 Deprecated: Array and string
```

offset access syntax with curly braces is deprecated in  
/homepages/0/MYHOME/htdocs/tiki-18.8/lib/tikilib.php on line 5779

Importance

4

Easy to solve?

5

Priority

20

Demonstrate Bug (Tiki 19+)

This bug has been demonstrated on [show2.tiki.org](http://show2.tiki.org)

Demonstrate Bug (older Tiki versions)

Please demonstrate your bug on [show.tikiwiki.org](http://show.tikiwiki.org)

Version: 18.x ▼

Ticket ID

7838

Created

Saturday 25 September, 2021 09:43:03 GMT-0000  
by hman

LastModif

Tuesday 02 November, 2021 11:16:30 GMT-0000

Comments



Jonny Bradley 27 Sep 21 11:16 GMT-0000

Sorry, 18.x only supports up to php 7.2.

See Requirements for more (and of course that needs updating i think ;)



hman 27 Sep 21 15:00 GMT-0000

Requirements states minimum requirements, not an upper limit for PHP...

I would think that 7.4 meets the requirement "minimum 7.2" stated there.

Btw., 7.2 is no longer free of charge. My ISP charges extra for having old versions running...




Marc Laporte 27 Sep 21 15:45 GMT-0000

<https://doctiki.org/Requirements>

[Load](#)
[Install](#)
[Backup](#)
[Upgrade](#)
[Help](#)

PHP



We are moving to a machine readable format:

[https://gitlab.com/tikiwiki/tiki-manager/-/blob/master/config/tiki\\_requirements.yml](https://gitlab.com/tikiwiki/tiki-manager/-/blob/master/config/tiki_requirements.yml)

Tiki Version	Minimum PHP Version	Maximum PHP Version	Notes
Tiki22 and Tiki23	7.4	Too soon to know	
Tiki21 LTS	7.2	Too soon to know	
Tiki19 or Tiki20	7.1	7.2 (maybe 7.3 as well)	There is known issue with Tiki 20.2 where PHP requirement was bumped up to PHP 7.2 accidentally
Tiki18 LTS	5.6	7.2	
Tiki15 LTS	5.5	5.6	
Tiki12 LTS	5.3	5.6	



hman 30 Sep 21 19:34 GMT-0000

This spells out to TWO current LTS that cannot be run on current PHP! PHP 7.3 (stated to be maximum for Tiki 21 LTS) is also EOL end of this year... I don't think that this is wise...



Marc Laporte 01 Oct 21 21:29 GMT-0000

Tiki 18.0 was released in January 2018, way before PHP 7.3.0 so it is not supposed to support it.

For each PHP version, there is at least 1 Tiki LTS supported.



hman>Marc Laporte 01 Oct 21 22:17 GMT-0000

Replied to

That may be technically correct, but irrelevant. Like it or not, IMHO Tiki must support the PHP versions of its service life. If you are on rented space, which I assume the vast majority is, there is no choice of PHP version. I am lucky my ISP does give that choice, but it costs a premium to run Tikis favourite.

Sorry, IMHO a sentence like "For each PHP version, there is at least 1 Tiki LTS supported" totally misses the point. When 7.3 goes EOL, there will be no PHP 7.3 for many people and that means: No Tiki LTS, because Tiki 21 doesn't support 7.4. Maybe the next LTS after 21 might do it, but it's not released yet.

Having said that I must add that the problems of 18.8 with 7.4 seem to be strictly limited to

the installer. Tiki itself runs on 7.4...



Marc Laporte>hman 02 Oct 21 03:16 GMT-0000

Replied to  
That may...

PHP 7.3 is supported until Dec 2021

Tiki 24 is planned for Feb 2022.

You are correct. There is a gap! If someone is forced to PHP 7.4, they also must upgrade to 23.x, and then to 24.x

FYI, for Tiki24, we are working towards supporting PHP 7.4, 8.0 and hopefully 8.1 The Tiki part is mostly done, and the challenges are dependencies like Smarty.  
<https://github.com/smarty-php/smarty/pull/629>



Marc Laporte>Marc Laporte 02 Oct 21 03:26 GMT-0000

Replied to  
PHP 7.3...

In recent times, PHP has a new version every year, and is supported for 3 years.

I hear your concerns. Now, how does this convert into a change of policy?

<https://tiki.org/Versions> contains the best I can imagine. I don't know how to do better but I'll be very happy if someone finds a better way of doing it.

Please keep in mind that some things are under our control (Tiki code) but others, not (Tiki dependencies). And we don't know in advance the scope of breaking changes. For example, when will be PHP 9.0? and how many breaking changes will it have? Tiki18 is good because it works with PHP 5.6, 7.0, 7.1 and 7.2, but will the transition to PHP 9.0 permit something similar?

Thanks!



hman>Marc Laporte 02 Oct 21 11:57 GMT-0000

Replied to  
In recent...

In my humble opinion (please bear with me, I do not want to offend anyone) Tiki must reduce its pace for new features. Actually, given the vast number of unfixed bugs, it is my belief Tiki needs a resting period to catch up with bugfixing. That would also reduce the stress of adaptation to new PHP releases.

As the user needs a working combo of Tiki, PHP, MySQL etc. I think it would be useful to overlay the lifetime charts of Tiki with that of PHP, Apache and at least MySQL (I do not have figures for that, but I imagine most Tikis will run on Apache and MySQL, because it is what you get "ready made" by ISPs when you book a (virtual or dockered) LAMP server.

That Tiki depends on third party vendors of course adds another layer of complexity. In my opinion, Tiki should try to strip as many of them as possible. Not just because of updating dependencies, but also because many of them are untranslated and untranslatable (!). Which, btw., is something you do not tell users... Untranslated/untranslatable plugins IMHO must be marked "English only" so admins know what they get and can make an informed decision.

Reduction of third party code is also highly advisable in terms of security updates! If you don't, then you might have to security update more often...



Marc Laporte>hman 02 Oct 21 18:29 GMT-0000

Replied to  
In my...

I am not offended. Everyone wants fewer bugs. But how?

"Tiki must": Tiki is not just one person. It is a diverse group with different (hopefully converging) objectives. To collaborate on goals/objectives (like we have done many times before), we need a proposal with guidelines we can agree on.

<https://tiki.org/Versions> explains how we do things, along with the pros and cons.  
<https://tiki.org/forumthread65674> is a consultation for a decision.

"A resting period to catch up with bugfixing": How does this work? A feature freeze? For those that code new features: what do we tell them? Keep them in a fork until date X?

"Tiki should try to strip as many of them as possible"

Ok, which ones? Who decides? Who does the work? And what do we tell people who are using them? We tell them that the features are removed after Tiki24LTS, so they can stay on that version for 5 years?

Related discussion: Rolling Release of Tiki



hman>Marc Laporte 03 Oct 21 12:21 GMT-0000

Replied to  
I am not...

""A resting period to catch up with bugfixing": How does this work? A feature freeze?"

Yes. Tiki's monolithic structure has had some advantages in the past, but in the present it makes improvement impractical (as well as installations - for 18.8 you have to transfer 28 thousand files, and not everyone has SSH access). The amount of unfixed bugs is so vast, it has to be dealt with. To be honest with you, I've been using Tiki since version 1.9.x. 18.8 is the worst I have ever run. So many basic stuff that just doesn't work (menus, calendars...), it made me stop advocating for Tiki.

"For those that code new features: what do we tell them? Keep them in a fork until date X?" I would think that moving that into real (!) plugins or add-ons is the best way. So those early implementers can do their work, but it would not add up to the pile of unfixed stuff in the core. They would have their own pile. Which would also be beneficial for those implementers, because they can update much more often and independently.

"Ok, which ones? Who decides? Who does the work? And what do we tell people who are using them? We tell them that the features are removed after Tiki24LTS, so they can stay on that version for 5 years?"

Tiki has removed features before. Migrating from Tiki 12.14 to 18.3 was a huge undertaking for me, because so many things changed. It's okay given the time span between the two. I am led to believe that a) new integrations should only be ok'd if they get a full set of tra(). IMHO that should be integral part of any integration process. b) All code that does not translate (lacks tra()) must be clearly labelled in order to warn admins. If you run a localized Tiki, the one thing you cannot do is activate a feature that comes English only... Worst case scenario would be that the new feature is integrated "seamless", but it's error messages aren't. Those functions/features should be, IMHO, be first to either adapt or be replaced.

Tiki breaks it's language promise with that many functions/features that cannot be translated!



hman>hman 03 Oct 21 12:45 GMT-0000

Replied to  
""A...

An indicator for this could be that Tiki once was the leader in installations. Today, where fewer and fewer people undertake a custom installation, people got used to preinstalls not only for the OS of their PC, but for web software as well.

I don't have an overview over the US market, but in Germany, when you rent webspace, you can get Wordpress or Joomla pre-installed, pre-configured and with support by the ISP. I know no ISP where you could have Tiki, unless you install it yourself.



Bernard Sfez / Tiki Specialist>hman 04 Oct 21 04:37 GMT-0000

Replied to

""A...

I just added a comment [here](#) before reading this discussion.

IMO;

"Features freeze" is not doable. Features are added because someone (a paying customer) is asking for it and we need developers and consultants making a decent living with Tiki so they stay on Tiki and don't leave. (we need to stop the bleeding)

However we may "Features prioritise" for test, fixes, improvements and monitoring. For exemple basic usage of Wiki page should be 99% free from issues and working like a charm (UX|UI) from the user point of view as this is a "Tiki core feature" and the first thing a user will start with (Install comes first that's true). Any addition/change (commit) should be check more carefully and tested with various configuration (not only with predogfood servers)



hman>Bernard Sfez / Tiki Specialist 04 Oct 21 19:18 GMT-0000

Replied to

I just...

With all due respect (I don't want to offend you):

In my humble opinion you are wrong.

If you "want to stop the bleeding" then make Tiki RUN. At the moment even the most basic stuff like creating menus or using calenders DOES NOT RUN.

I am ISTQB certified professional software tester. I know no product where new features would be coded when core features aren't working. If they have showstopper errors, the show is being stopped and that means that 100% of manpower goes into bug fixing, possibly even exceeding 100% depending on impact analysis.



Bernard Sfez / Tiki Specialist>hman 05 Oct 21 06:48 GMT-0000

Replied to

With all...

All good, we are discussing and divergence of opinion is healthy. ;-)

But I don't understand why you say I'm wrong and wrote this in the same post:

hman wrote on 04 Oct 21:

*At the moment even the most basic stuff like creating menus or using calenders DOES NOT RUN*

Exactly my point.

Basic stuff should RUN.

What is considered basic stuff can be discussed, but basic stuff should be a priority.

And the rest of your post says as I do "Core feature should work".

From here, and we may think differently, I don't see the need for a TOTAL freeze of Tiki. But more a prioritisation in testing, monitoring and fixing.

1. Core features should work 99% and be "protected" (how and what is a sub-discussion)
2. New features (needed for some projects) can live their life as long as they are useful for Tiki, doesn't perturb core features and are optionals.

Shutting down new features development in Tiki won't get more resources to fix core features. Tiki is not a company with developers or consultants waiting to complete tasks during their fixed working hours.

The developers add new features because they needed it or a customer needs it. One way or another they do it to put money on their table. While they do so they also usually fix or clean other part of the code they are reaching during their work. It is very very unlikely, they'll use a potentially "freed" time to fix bugs they don't feel concerned with in Tiki.

Overall I see more convergence than differences between us and I hope my post clarify. ;-)



hman>Bernard Sfez / Tiki Specialist 05 Oct 21 22:09 GMT-0000

Replied to

All good,...

"Features freeze" is not doable."

In my opinion, it's the only thing that can save Tiki! Tiki has grown WAY out of proportion. It's so big that the shrinking developer numbers can no longer maintain it. The complexity is too much even for devs. Bugs are piling up instead of going down. Adding even more complexity IMHO makes things worse, unless Tiki development can be detached from plugin development, and with that I mean real plugins that won't be preinstalled and can be added (and safely removed!) later by admins wishing to gain those features. They would have their own version numbers, own life cycles etc.

Sure, Tiki is not a company, so we cannot do resource management with standard methods. But frankly, why should Tiki care about developers who only want to serve short to mid-term customer demands while not caring about the bigger picture? Those devs who don't see that they, too, rely on a stable core where their code can run on. So it should be in their own best interest to make (and keep) Tiki run. And at some time, this is going to frustrate the customers paying those devs.

If such devs just want to drop their code into Tiki and not make after care for their own code, IMHO we should, or rather must, draw a strict line of separation to make sure such code can be removed if it proves troublesome at a later point in time.

Look at the Linux development model. Lots and lots of companies contribute, but Linus Torvalds is VERY selective at what he allows to enter the kernel. He favors long time benefits for all over short term (singular) customer satisfaction.





Marc Laporte>hman 07 Oct 21 04:29 GMT-0000

Replied to  
"Features..."

**@hman:** now it's my turn to hope to not offend you. You may have software testing experience, but do you have experience in successfully managing a large Open Source project?

I do.

Please indulge me for a moment: Let's start with something simple and practical, succeed, and build on that.

You write: "new integrations should only be ok'd if they get a full set of tra(). IMHO that should be integral part of any integration process. b) All code that does not translate (lacks tra()) must be clearly labelled in order to warn admins. If you run a localized Tiki, the one thing you cannot do is activate a feature that comes English only... "

1- Monitoring takes time. Who will monitor? Who will enforce?

2- Labelling code takes time. Who will do?

3- For those that work in English: They don't see the issue, so it's not an obvious contribution. They are not doing this to cause issues. There is simply no problem for them.

Let's start with something really simple. When a tra() is missing, just add it. No need to make a bug report. Just add it. It takes about 1 minute to add a tra(). You can make merge requests directly via the GitLab web interface. No need for a complicated setup. It's very much like editing a wiki page. Elsewhere on dev.tiki.org, you have been encouraged to contribute fixes directly. I'd like to reinforce that this is what I hope you will do. If you need help to contribute, we will coach you.

Please do everything on trunk (master) for now so it will be in Tiki24 for sure. We will quickly review them and merge them in. And once we have this process working smoothly , we can evaluate backports to previous versions.

Best regards,



hman>Marc Laporte 22 Oct 21 13:49 GMT-0000

Replied to

"1- Monitoring takes time. Who will monitor? Who will enforce?" Automate that...

"2- Labelling code takes time. Who will do?"

Can be EASILY automated. Code without tra() is English only... Or simply make it a default, so ALL new code will be labelled in this fashion, and developers could be informed that in order to have that label removed from their functions THEY have to supply tra() function

calling.

The window of opportunity that short term work had provided has closed. I used it to implement a web presence for a non-profit organisation. Work had begun on Tiki 18 LTS and it went into productive status on 18.8.

Migration to 21 LTS can be begun no sooner than 2022. After the very negative experience with Tiki 18 (calendar mostly non-functional, menus definitely non-functional etc. pp. et. al) no version advancements will be made before functionality has been reached. This is mandatory, as a fallback to a functional version is of the essence, if version advancement results in new bugs.

Only LTS versions are acceptable.



Marc Laporte>hman 28 Oct 21 02:10 GMT-0000

Replied to

"1-... @hman: Please see: How to Have Fewer Bugs in Tiki



Marc Laporte>hman 28 Oct 21 02:23 GMT-0000

Replied to

"1-...

@hman: How much work is it (approx) for you to do a test upgrade to 21.x and test (at 80% confidence level) that everything is OK?

Tiki 18 menus: I agree they are bad. Explained at How to Have Fewer Bugs in Tiki

Tiki 18 calendars: I don't remember. But I do know that FullCalendar on vs off had a big impact. We later got rid of legacy interface and made FullCalendar the default. By converging our energies on common code, we do better.

I use Tiki 21x massively and it is generally very very good. and we fix all bugs that are in our way. Menus in 21x are better than 18x but still some issues. Calendars in 21x: I don't use enough to say.



hman>Marc Laporte 31 Oct 21 16:46 GMT-0000

Replied to

How much work is it (approx) for you to do a test upgrade to 21.x and test (at 80% confidence level) that everything is OK?

A month, at the minimum. Time that I don't have. Even a simple upload of Tiki 21.4, which is

TWICE the size of 18.8 takes two hours...

Tiki 18 menus: I agree they are bad.

No. They are not bad. They don't work at all. It is not a drag & drop problem: You simply cannot create them from scratch!!

Tiki 18 calendars:

The number of bugs is just staggering. From double created events to un-editable recurring events to simply broken DIV and CSS, it can only be called a disaster.

Also, i just do not understand why they don't have the feature of a recurring event? It pretends to have that, but in fact, it doesn't. It simply makes n copies of the event. If you make an event recur 100 times, you get 100 events. Imagine the amount of work if you want to correct a simple typo in them...



hman>hman 31 Oct 21 18:32 GMT-0000

Replied to  
How much...

As it turns out, after those two hours of upload (I test-upgraded the backup test site, not the productive one) Tiki won't start.

The reason is that only the productive site has it's own domain, while the test & backup site sits on my personal domain.

And that is still on PHP 5.6, because there is my trusty Tiki 12.14...

I cannot switch PHP for both independent from each other, so a first I must migrate 12.14 to 18.8 before I can switch my personal domain to PHP 7.4 or (and 7.2 or 7.3 take longer, because I have to sign a support contract for that (the current one is for 5.,6)



hman>hman 31 Oct 21 23:23 GMT-0000

Replied to  
As it...

SHIT! Forgive the profanity, but upgrading the 12.14 to 18.8 left me without the possibility to log in as admin. And the sentence in the upgrade wizard that tells what to do in that case does not show up again...

Upgrading itself did not produce visible problems, I just cannot login as admin to unlock the site...



Torsten Fabricius>hman 01 Nov 21 01:18 GMT-0000

Replied to  
SHIT!...

Hi hman,  
in this case you can google for example for  
"Tiki reset lost my admin password"

and you get this:

<https://doc.tiki.org/Lost-admin-password>

In case you have no access or not setup the password reset in the login box/popup.

Just to let you know.

I was happy quite often, that someone in the Tiki community did write this wiki page. Helped me a lot of times.

Regards,  
Torsten



hman>Torsten Fabricius 01 Nov 21 14:58 GMT-0000

Replied to  
Hi hman,...

Thank you. I could help myself be restoring the damaged password from backup. But something else happened: The upgrade of the other Tiki, from 18.8 to 21.4 had one SQL request failing:



```
UPDATE `users_objectpermissions` SET `permName` = 'tiki_p_read_article' WHERE  
`permName` = 'tiki_p_topic_read' AND `objectType` = 'topic' Duplicate entry  
'4dadd7eebd7a5fc8790114f9f5aaa8b6-topic-Anonymous-tiki_p_read_art' for key  
'PRIMARY'
```



hman>hman 01 Nov 21 15:08 GMT-0000

Replied to  
Thank...

For the moment, I checkmarked this and continued. The result is a complete layout disaster. My classic Tiki layout (3 module zones and a header) is gone. What has been the left module zone is now on the right-hand side, where TWO module zones are. So left changed into right, and right changed into far-right. The control panels are now one single column on a seemingly endless page, hardly usable at all.

Next nail into Tikis coffin.

The menu is a mess with wrong fonts, wrong sizing, wrong spacing etc.  
Conversion is just as bad as 12->18 has been.



hman>Marc Laporte 22 Oct 21 14:08 GMT-0000

Replied to

I currently experience an issue that could be PHP version related. I wrote before that only the installer for Tiki 18.8. has issues on PHP 7.4. It is possible that this is not entirely true. Today I realized I cannot display or edit a tracker:



---

```
Fatal error: Uncaught Error: Cannot pass parameter 2 by reference in  
MYHOME/tiki-18.8/tiki-view_tracker.php:529 Stack trace: #0 {main} thrown in  
MYHOME/tiki-18.8/tiki-view_tracker.php on line 529
```

This is the underlying reason why I get user complaints that they cannot update the shop system with new products. They are users, so they simply get a clean white page containing nothing (at all). As admin, I get this fatal error. I have yet to analyze if this still happens when I downgrade the PHP version (in the hope it can be downgraded at all. The ISP discourages downgrading).



hman>hman 22 Oct 21 14:14 GMT-0000

Replied to

I...

line 529 is



---

```
$smarty->assign_by_ref('fields_count', count($xfields['data']));
```

Strangely, PHP does not complain about the numerous other assign\_by\_ref before line 529...



hman>hman 22 Oct 21 14:49 GMT-0000

Replied to

I...

Confirmed, a direct incompatibility with PHP 7.4. Downgrading to 7.3 is a (temporal) workaround for the issue.

Only until 12/2021, after that my ISP demands making an extended support contract and paying a premium, in order to keep 7.3.

Strangely, what leads to a fatal error on 7.4 does not even produce a "deprecated" warning

while running on 7.3...



Marc Laporte>hman 28 Oct 21 02:29 GMT-0000

Replied to  
I...

I recommend that you not run Tiki in unsupported PHP versions.

You are wasting your time and other people's time that are reading this.

Tiki 18x is max PHP 7.2



hman>Marc Laporte 31 Oct 21 17:01 GMT-0000

Replied to  
I...

Then I will repeat that in my view, any (!) LTS release that claims to be currently supported MUST run on the PHP version of it's service time. Tiki 18.8. does not.

Also, Tiki installer MUST issue warnings of incompatibility for the detected PHP, and in the control center, warning should be issued if the currently detected PHP is under minimum or above maximum.



hman>hman 31 Oct 21 19:19 GMT-0000

Replied to

Then I... Side note: My old 12.14 still says that it would be continued to be supported, in the Control Panels... I know that it isn't, but there is something that makes it think it is...



Torsten Fabricius>hman 01 Nov 21 01:11 GMT-0000

Replied to  
Then I...

No hman,  
an LTS has to be guaranteed to run on the same machines and programming languages it was designed for and from the same period of time.

Thus Tiki 18.x shall not be updated to PHP 7.4 or 8.0 and therefore break down running

websites on still 5.6/7.2 webserver after a regular SVN or GIT update. This would be an insane failure by concept.

Once you need or want or should upgrade your PHP you shall upgrade your web applications and once you upgrade your web applications you shall upgrade your webserver. This is standart procedure. And if you need to run old besides state of the art web applications, you need to pay extra fees to your provider.

I do not get your problem, sorry.

If you want to go for PHP 7.4, which makes sense, then go for Tiki 21.x LTS, which should be quite fast and easy. This is why we have a 3 semester (1 1/2 years) or 3 versions LTS release circle, for that you have the option to stay up to date server wise and still stay in LTS .

From my point of view this makes sense!

Torsten



hman>Torsten Fabricius 01 Nov 21 16:01 GMT-0000

Replied to  
No hman,...

*an LTS has to be guaranteed to run on the same machines and programming languages it was designed for and from the same period of time.*

*Thus Tiki 18.x shall not be updated to PHP 7.4 or 8.0 and therefore break down running websites on still 5.6/7.2 webserver after a regular SVN or GIT update. This would be an insane failure by concept.*

Please explain how an upgrade that enables PHP 7.4 "breaks" when running on 7.2...

With 5.6 it is agreedly a little more complicated, but my statement is "must support all PHP of it's service time", and that leads to 18.8 NOT breaking on 5.6... Yes, this makes coding a little harder, it might introduce the necessity of code level branches that everyone hates. But, please, send kudos to the PHP developers. THEY are responsible for the deplorable fact that PHP breaks it's own rules even on minor releases. If not even developers from bigger projects like Tiki speak up, no one will stop them breaking everyone's code once again.

Ten years ago I thought (then) Sun's Java would be wild with constant breaking of code, but it's harmless compared to what they've done to PHP from 5.6 -> 8.

*Once you need or want or should upgrade your PHP you shall upgrade your web applications and once you upgrade your web applications you shall upgrade your webserver. This is standart procedure.*

No. I choose an LTS to be spared constant updates, in order to receive plannable (!) service life. When I have to update because BOTH PHP and Tiki can force that, the whole concept of LTS breaks and service life gets seriously and unplannably shortened. It is NOT possible to "stay up to date server wise and still stay in LTS". In order to do so. Tiki 18.8 must at least tolerate PHP 7.4 which becomes EOL at about the same time.


To me, this is unacceptable. It is already very hard to explain users why every couple of years the tedious task of release change has to be undergone, with all risk looming like breaking the layout, functionality, etc. But even shortening that to one to two years, I know no user who would accept that and produce content that he or she might lose or see defaced in not so far future. And I do not talk about arcane, complicated features with ultra-complex data structures. I'm talking about Wiki content, menus and calendars that do not migrate easily.



Marc Laporte 27 Sep 21 15:48 GMT-0000

Why your ISP charges extra: Extra work because PHP 7.2 is no longer supported by PHP:  
<https://www.php.net/eol.php>

#### Attachments

filename	created	hits	comment	version	filetype
 2021-09-27_114329.png	27 Sep 21 15:44 GMT-0000	0			

The original document is available at  
<https://dev.tiki.org/item7838-Installer-not-ready-for-PHP-7-4-in-Tiki-18-x>