

To Mods Or Not To Mods

To Mods Or Not To Mods

~~Please help improve this page with factors for/against have certain code in mods.~~

Overview

As of end of 2017 / early 2018 the new Feature **Packages** was implemented in Tiki and the mods feature is finally getting to be retired. Thus this page is mainly staying for historic reasons.

Please follow up here for the packages feature:

- [Tiki Packages vs built-in](#)
- [Packages](#)
- [Composer Dependencies Revamp](#)

Here a number of links regarding the story behind **Mods**:

- <https://tiki.org/forumthread67983-About-Tiki-Packages-in-Tiki-18-0-LTS-and-why-I-support-this-project-And-mods-tiki-org-can-now-be-scheduled-for-retirement>
- <https://sourceforge.net/p/tikiwiki/mailman/message/13455128/>
- <http://tikiwiki.1073613.n5.nabble.com/The-future-of-Mods-td18857.html>
- <http://tikiwiki.1073613.n5.nabble.com/The-state-of-mods-Cleanup-of-mods-where-are-we-going-with-th-is-tp8110p8112.html>
- <http://tikiwiki.1073613.n5.nabble.com/Moving-mods-goodies-to-main-codebase-especially-plugins-td16074.html>
- <https://tiki.org/Webinar-2012-10> (Go to minute 82)

---

In systems like Drupal or Joomla!, most of the available functionality is available through 3rd party modules, extensions, etc (different systems have different names). You need to install separately. In TikiWiki, it's the opposite. Most of the functionality is built-in and you decide to activate or not. Both approaches have their pros & cons. You can read more about the Tiki **Model** and <http://pluginproblems.com/>

As with many things, it's not always black & white. There are many shades of gray. The criteria below is intended as a guide to help developers decide is something should be "core" or "mods". Please improve/debate these!

License

If code is not compatible with LGPL (see:[LibLicense](#)), ask for relicense/dual license. If not, goto Mods.

Usage/popularity

Popular stuff should be in main code base. Very specific things in mods.

Size/complexity of code

Large external libs? Do we really want to add several more megs to the package?

But if it's a few dozen lines of code, is it really worth the overhead of the mods installer?

## Stability / Is it maintained?

- If a feature is not yet ready, mods can be a good place to experiment. Once it goes into core, people expect that there will be a clean upgrade path (You can also put in core, and tag as experimental). See next point about DB.
- If a feature is buggy, it can better in mods. However, features tend to improve **if** they are used and people report bugs, contribute fixes, etc.
- If a feature is being phased out, mods can be a place so it's still available to the very small number of people that still need it. Once people have data in a feature however, it's very difficult to phase out. See next point about DB.

## Does it involve affecting the database?

Does your code make many changes to data in core db tables? This could be a sign (though not necessarily) that it should be part of core rather than mods. Icon packs, themes and most wiki-plugins don't affect the database. So mods is OK.

## Level of version dependency

Please see [Dependency hell](#). If code will work for sure in 1.9.x and 2.x, or so long as it is clear that you depend on a certain version, Mods is fine.

## Reliance on 3rd party code, available to shared hosting

Something which depends on a piece of software which is typically not found on shared hosting should generally go to mods.

## Translation

Things in mods tend not to get translated.

## Security

If something can be potentially unsafe, better to send to mods.

So, again, there is no black or white. Just use common sense and consult with others. Some things like Tiki Maps depend on Mapserver, which is not available on shared hosting. On the other hand, it is maintained by Franck and he prefers it be in the core. So Tiki Maps is part of the core. Also mods did not exist when Tiki Maps was created so it may move in mods eventually.