EXHIBIT 13

Detailed Plugin Guidelines

In this article

1. Plugins must be compatible with the GNU General Public License
2. Developers are responsible for the contents and actions of their plugins.
3. A stable version of a plugin must be available from its WordPress Plugin Directory page.
4. Code must be (mostly) human readable.
5. Trialware is not permitted.
6. Software as a Service is permitted.
7. Plugins may not track users without their consent.
8. Plugins may not send executable code via third-party systems.
9. Developers and their plugins must not do anything illegal, dishonest, or morally offensive.
10. Plugins may not embed external links or credits on the public site without explicitly asking the user's permission.
11. Plugins should not hijack the admin dashboard.
12. Public facing pages on WordPress.org (readmes) must not spam.
13. Plugins must use WordPress' default libraries.
14. Frequent commits to a plugin should be avoided.
15. Plugin version numbers must be incremented for each new release.
16. A complete plugin must be available at the time of submission.
17. Plugins must respect trademarks, copyrights, and project names.
18. We reserve the right to maintain the Plugin Directory to the best of our ability.

Last Updated: March 15, 2024

Adding a Block Only plugin? Please read the Block Specific Guidelines

The Plugin Directory

The goal of the WordPress Plugin Directory is to provide a safe place for all WordPress users – from the non-technical to the developer – to download plugins that are consistent with the goals of the WordPress project.

To that end, we want to ensure a simple and transparent process for developers to submit plugins for the directory. As part of our ongoing efforts to make the plugin directory inclusion process more transparent, we have created a list of developer guidelines. We strive to create a level playing field for all developers.

If you have suggestions to improve the guidelines, or questions about them, please email `plugins@wordpress.org` and let us know.

Developer Expectations

Developers, all users with commit access, and all users who officially support a plugin are expected to abide by the following guidelines:

- Plugin Directory Guidelines (this document)
- Community Guidelines
- Forums Guidelines (should they use the forums/reviews)

Violations may result in plugins or plugin data (for previously approved plugins) being removed from the directory until the issues are resolved. Plugin data, such as user reviews and code, may not be restored depending on the nature of the violation and the results of a peer-review of the situation. Repeat violations may result in all the author's plugins being removed and the developer being banned from hosting plugins on WordPress.org.

It is the responsibility of the plugin developer to ensure their contact information on WordPress.org is up to date and accurate, in order that they receive all notifications from the plugins team. Auto-replies and emails that route to a support system are not permitted as they historically prevent humans from addressing emails in a timely fashion.

All code in the directory should be made as secure as possible. Security is the ultimate responsibility of the plugin developer, and the Plugin Directory enforces this to the best of our ability. Should a plugin be found to have security issues, it will be closed until the situation is resolved. In extreme cases the plugin may be updated by the WordPress Security team and propagated for the safety of the general public.

While we attempt to account for as many relevant interpretations of the guidelines as possible, it is unreasonable to expect that every circumstance will be explicitly covered. If you are uncertain whether a plugin might violate the guidelines, please contact `plugins@wordpress.org` and ask.

The Guidelines

1. Plugins must be compatible with the GNU General Public License

Although any GPL-compatible license is acceptable, using the same license as WordPress — "GPLv2 or later" — is strongly recommended. All code, data, and images — anything stored in the plugin directory hosted on WordPress.org — must comply with the GPL or a GPL-Compatible license. Included third-party libraries, code, images, or otherwise, must be compatible. For a specific list of compatible licenses, please read the GPL-Compatible license list on gnu.org.

2. Developers are responsible for the contents and actions of their plugins.

It is the sole responsibility of plugin developers to ensure all files within their plugins comply with the guidelines. Intentionally writing code to circumvent guidelines, or restoring code they were asked to remove, is prohibited (see #9 illegal/dishonest actions).

Developers are expected to confirm, before uploading to SVN, the licensing of all included files, from original source code to images and libraries. In addition, they must comply to the terms of use for all third party services and APIs utilized by their plugins. If there is no way to validate the licensing of a library or the terms of an API, then they cannot be used.

3. A stable version of a plugin must be available from its WordPress Plugin Directory page.

The only version of the plugin that WordPress.org distributes is the one in the directory. Though people may develop their code somewhere else, users will be downloading from the directory, not the development environment.

Distributing code via alternate methods, while not keeping the code hosted here up to date, may result in a plugin being removed.

4. Code must be (mostly) human readable.

Obscuring code by hiding it with techniques or systems similar to `p,a,c,k,e,r`'s obfuscate feature, uglify's mangle, or unclear naming conventions such as `$z12sdf813d`, is not permitted in the directory. Making code non-human readable forces future developers to face an unnecessary hurdle, as well as being a common vector for hidden, malicious code.

We require developers to provide public, maintained access to their source code and any build tools in one of the following ways:

- Include the source code in the deployed plugin
- A link in the readme to the development location

We strongly recommend you document how any development tools are to be used.

5. Trialware is not permitted.

Plugins may not contain functionality that is restricted or locked, only to be made available by payment or upgrade. Functionality may not be disabled after a trial period or quota is met. In addition, plugins that provide sandbox only access to APIs and services are also trial, or test, plugins and not permitted.

Paid functionality in services *is* permitted (see guideline 6: serviceware), provided all the code inside a plugin is fully available. We recommend the use of add-on plugins, hosted outside of WordPress.org, in order to exclude the premium code. Situations where a plugin is intended as a developer tool only will be reviewed on a case by case basis.

Attempting to upsell the user on ad-hoc products and features *is* acceptable, provided it falls within bounds of guideline 11 (hijacking the admin experience).

6. Software as a Service is permitted.

Plugins that act as an interface to some external third party service (e.g. a video hosting site) are allowed, even for paid services. The service itself must provide functionality of substance and be clearly documented in the readme file submitted with the plugin, preferably with a link to the service's Terms of Use.

Services and functionality *not* allowed include:

- A service that exists for the sole purpose of validating licenses or keys while all functional aspects of the plugin are included locally is not permitted.
- Creation of a service by moving arbitrary code out of the plugin so that the service may falsely appear to provide supplemented functionality is prohibited.
- Storefronts that are not services. A plugin that acts only as a front-end for products to be purchased from external systems will not be accepted.

7. Plugins may not track users without their consent.

In the interest of protecting user privacy, plugins may not contact external servers without *explicit* and authorized consent. This is commonly done via an 'opt in' method, requiring registration with a service or a checkbox within the plugin settings. Documentation on how any user data is collected, and used, should be included in the plugin's readme, preferably with a clearly stated privacy policy.

Some examples of prohibited tracking include:

- Automated collection of user data without explicit confirmation from the user.
- Intentionally misleading users into submitting information as a requirement for use of the plugin itself.
- Offloading assets (including images and scripts) that are unrelated to a service.
- Undocumented (or poorly documented) use of external data (such as blocklists).
- Third-party advertisement mechanisms which track usage and/or views.

An exception to this policy is Software as a Service, such as Twitter, an Amazon CDN plugin, or Akismet. By installing, activating, registering, and configuring plugins that utilize those services, consent is granted for those systems.

8. Plugins may not send executable code via third-party systems.

Externally loading code from documented services is permitted, however all communication must be made as securely as possible. Executing outside code within a plugin when not acting as a service is not allowed, for example:

- Serving updates or otherwise installing plugins, themes, or add-ons from servers other than WordPress.org's
- Installing premium versions of the same plugin
- Calling third party CDNs for reasons other than font inclusions; all non-service related JavaScript and CSS must be included locally
- Using third party services to manage regularly updated lists of data, when not explicitly permitted in the service's terms of use
- Using iframes to connect admin pages; APIs should be used to minimize security risks

Management services that interact with and push software down to a site *are* permitted, provided the service handles the interaction on it's own domain and not within the WordPress dashboard.

9. Developers and their plugins must not do anything illegal, dishonest, or morally offensive.

While this is subjective and rather broad, the intent is to prevent plugins, developers, and companies from abusing the freedoms and rights of end users as well as other plugin developers.

This includes (but is not restricted to) the following examples:

- Artificially manipulating search results via keyword stuffing, black hat SEO, or otherwise
- Offering to drive more traffic to sites that use the plugin
- Compensating, misleading, pressuring, extorting, or blackmailing others for reviews or support
- Implying users must pay to unlock included features
- Creating accounts to generate fake reviews or support tickets (i.e. sockpuppeting)
- Taking other developers' plugins and presenting them as original work
- implying that a plugin can create, provide, automate, or guarantee legal compliance
- Utilizing the user's server or resources without permission, such as part of a botnet or crypto-mining
- Violations of the WordPress.org Community Code of Conduct
- Violations of the WordCamp code of conduct
- Violations of the Forum Guidelines
- Harassment, threats, or abuse directed at any other member of the WordPress community
- Falsifying personal information to intentionally disguise identities and avoid sanctions for previous infractions
- Intentionally attempting to exploit loopholes in the guidelines

10. Plugins may not embed external links or credits on the public site without explicitly asking the user's permission.

All "Powered By" or credit displays and links included in the plugin code must be optional and default to *not* show on users' front-facing websites. Users must opt-in to displaying any and all credits and links via clearly stated and understandable choices, not buried in the terms of use or documentation. Plugins may not require credit or links be displayed in order to function. Services *are* permitted to brand their output as they see fit, provided the code is handled in the service and not the plugin.

11. Plugins should not hijack the admin dashboard.

Users prefer and expect plugins to feel like part of WordPress. Constant nags and overwhelming the admin dashboard with unnecessary alerts detract from this experience.

Upgrade prompts, notices, alerts, and the like must be limited in scope and used sparingly, be that contextually or only on the plugin's setting page. Site wide notices or embedded dashboard widgets *must* be dismissible or self-dismiss when resolved. Error messages and alerts must include information on how to resolve the situation, and remove themselves when completed.

Advertising within the WordPress dashboard should be avoided, as it is generally ineffective. Users normally only visit settings pages when they're trying to solve a problem. Making it harder to use a plugin does not generally encourage a good review, and we recommend limiting any ads placed therein. Remember: tracking

referrals via those ads is not permitted (see guideline 7) and most third-party systems do not permit back-end advertisements. Abusing the guidelines of an advertising system will result in developers being reported upstream.

Developers are welcome and encouraged to include links to their own sites or social networks, as well as locally (within the plugin) including images to enhance that experience.

12. Public facing pages on WordPress.org (readmes) must not spam.

Public facing pages, including readmes and translation files, may not be used to spam. Spammy behavior includes (but is not limited to) unnecessary affiliate links, tags to competitors plugins, use of over 5 tags total, blackhat SEO, and keyword stuffing.

Links to directly required products, such as themes or other plugins required for the plugin's use, are permitted within moderation. Similarly, related products may be used in tags but not competitors. If a plugin is a WooCommerce extension, it may use the tag 'woocommerce.' However if the plugin is an alternative to Akismet, it may not use that term as a tag. Repetitive use of a tag or specific term is considered to be keyword stuffing, and is not permitted.

Readmes are to be written for people, not bots.

In all cases, affiliate links must be disclosed and must directly link to the affiliate service, not a redirect or cloaked URL.

13. Plugins must use WordPress' default libraries.

WordPress includes a number of useful libraries, such as jQuery, Atom Lib, SimplePie, PHPMailer, PHPass, and more. For security and stability reasons plugins may not include those libraries in their own code. Instead plugins must use the versions of those libraries packaged with WordPress.

For a list of all javascript libraries included in WordPress, please review Default Scripts Included and Registered by WordPress.

14. Frequent commits to a plugin should be avoided.

The SVN repository is a release repository, not a development one. All commits, code or readme files, will trigger a regeneration of the zip files associated with the plugin, so only code that is ready for deployment (be that a stable release, beta, or RC) should be pushed to SVN. Including a descriptive and informative message with each commit is strongly recommended. Frequent 'trash' commit messages like 'update' or 'cleanup' makes it hard for

others to follow changes. Multiple, rapid-fire commits that only tweak minor aspects of the plugin (including the readme) cause undue strain on the system and can be seen as gaming Recently Updated lists.

An exception to this is when readme files are updated solely to indicate support of the latest release of WordPress.

15. Plugin version numbers must be incremented for each new release.

Users are only alerted to updates when the plugin version is increased. The trunk readme.txt must always reflect the current version of the plugin. For more information on tagging, please read our SVN directions on tagging and how the readme.txt works.

16. A complete plugin must be available at the time of submission.

All plugins are examined prior to approval, which is why a zip file is required. Names cannot be “reserved” for future use or to protect brands (see #17: respect brands). Directory names for approved plugins that are not used may be given to other developers.

17. Plugins must respect trademarks, copyrights, and project names.

The use of trademarks or other projects as the sole or initial term of a plugin slug is prohibited unless proof of legal ownership/representation can be confirmed. For example, the WordPress Foundation has trademarked the term “WordPress” and it is a violation to use “wordpress” in a domain name. This policy extends to plugin slugs, and we will not permit a slug to begin with another product’s term.

For example only employees of Super Sandbox should use the slug “super-sandbox,” or their brand in a context such as “Super Sandbox Dancing Sloths.” Non-employees should use a format such as “Dancing Sloths for Superbox” instead to avoid potentially misleading users into believing the plugin was developed by Super Sandbox. Similarly, if you don’t represent the “MellowYellowSandbox.js” project, it’s inappropriate to use that as the name of your plugin.

Original branding is recommended as it not only helps to avoid confusion, but is more memorable to the user.

18. We reserve the right to maintain the Plugin Directory to the best of our ability.

Our intent is to enforce these guidelines with as much fairness as humanly possible. We do this to ensure overall plugin quality and the safety of their users. To that end, we reserve the following rights:

- ... to update these guidelines at any time.
- ... to disable or remove any plugin from the directory, even for reasons not explicitly covered by the guidelines.
- ... to grant exceptions and allow developers time to address issues, even security related.
- ... to remove developer access to a plugin in lieu of a new, active, developer.
- ... to make changes to a plugin, without developer consent, in the interest of public safety.

In return, we promise to use those rights sparingly and with as much respect as possible for both end users and developers.

First published
April 9, 2015

Last updated
March 15, 2024

Previous
Compliance Disclaimers

Next
How Your Plugin Assets Work