

Edit Manage Stats



Ingo Steinke

Posted on 9 Dec 2021

Don't Update WordPress Plugins ...

[#wordpress](#) [#php](#) [#watercooler](#) [#webdev](#)

... unless you know what you are doing!

Don't get me wrong: please **do update your plugins!**

Keep your system up to date! This includes your server, PHP version, WordPress core version, themes and plugins. But be careful how to do it once your setup has become seriously outdated!

How I destroyed (and recovered) my customer's WordPress website with a few clicks:

As web developers, we might get used to the safety of modern software development. Static code analysis, syntax highlighting, auto completion, automated regression tests and peer reviews on staging systems as well as the possibility to revert changes and deploy the previous working version not only help us to deliver stable quality software, but also spare us the thrill and excitement of editing production systems five minutes before a deadline.

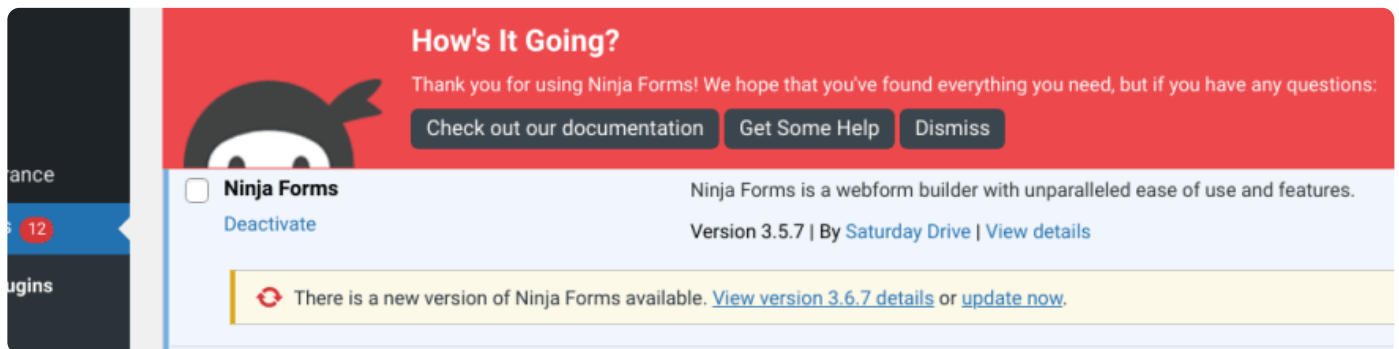
Unless it's WordPress.

Has it been tested to work with your setup?

"Works for me" does not mean that it will work for you, right? Even more so when it comes to ...

Updating WordPress Plugins 🤖

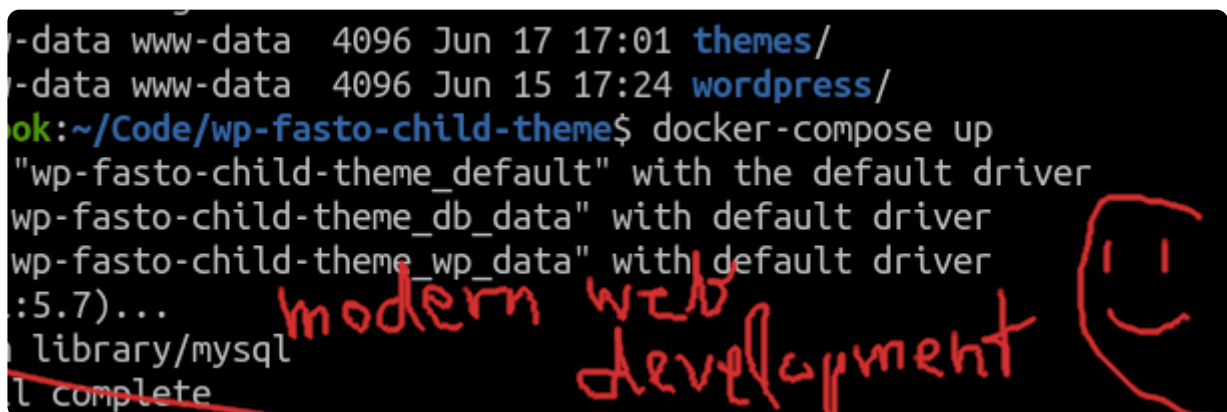
WordPress might be the most ubiquitous legacy software that still makes its users do edits and configuration changes directly on a single node production system.



WordPress core might not be a problem these days. But WordPress can be extended using themes and plugins, and anyone can upload mostly anything to the official plugin store to release code that has the power to override and break anything, causing fatal errors that make frontend and backend unavailable, unless you got the recovery email.

Works for me

In theory, as a web developer, I would develop a (child) theme on my local computer, test everything locally as well as on my customer's staging environment before deploying the tested version to the production system.



Assuming that there is a staging environment using the same system configuration, plugins and content as the production system.

In practice, my customer uses a page builder and tweaking some custom style sheets

In practice, my customer uses a page builder, and tweaking some custom style sheets is the least of the problems.

Outdated WordPress Plugins

I don't want to shame plugin developers. Ninja Forms is a useful no-code form builder that does its job and has more than a million active installations, according to its [official plugin page](#).

Preventing Fatal Errors

I don't want to shame WordPress either.

The community is very supportive, and the core developers have made a great effort to make the current versions quite stable and secure, at least compared to the situation about 10 years ago.

WordPress core tries to prevent installing and activating incompatible plugins, showing an error message and retaining the current plugin version.

But sometimes this does not work.

While Ninja Forms has done a good job staying backward compatible in the past, the latest versions have a problem with outdated PHP versions like PHP 7.1.

And after accepting the suggested minor update, WordPress was gone. No more links and menus in the administration, no more content in the frontend.

No more Content in the Frontend! 🤖

Instead, this:

```
Fatal error: Uncaught TypeError: Return value of
NF_Admin_Metaboxes_Calculations::registerReactMetabox() must be an instance
of void, none returned.
```

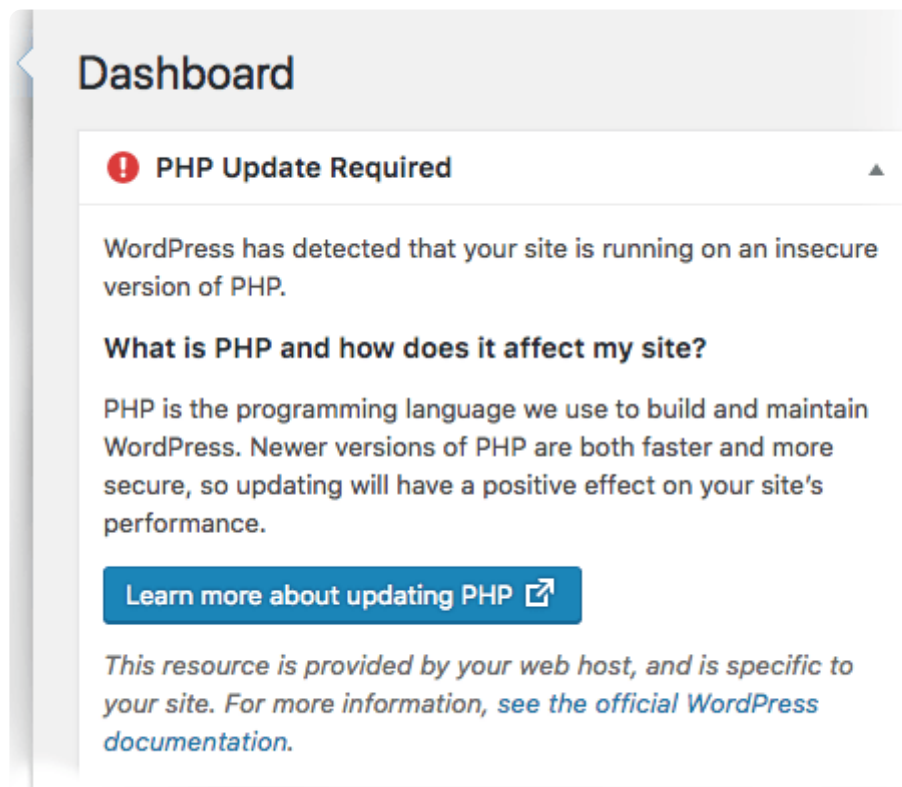
```
Fatal error: Uncaught TypeError: Return value of NF_Admin_Metaboxes_Calculations::registerReactMetabox() must be an instance of void, none returned
```

Fatal Error!!!

So something broke, because of some reason I won't be able to fix anyway, and WordPress makes it possible to use React in the backend and jQuery in the frontend at the same time in 2021.

As I did not have access to the mailbox where WordPress helpfully sent a recovery link (see below). I was hoping that maybe a PHP update might bring back the blog content.

Check Your Actual PHP Version



"PHP Update Required - WordPress has detected that your site is running on an insecure version of PHP." (But we won't tell you any more details here, as PHP 7.3 is just as bad as 5.3 ? Probably not!)

"What is PHP and how does it affect my site? ... Learn more about updating PHP", well who will click this button?

Either I didn't have any idea that PHP existed after all and I just want my blog to work so that I can post content and make it look good, or else I am a developer and think I know enough not to follow the link.

But do I actually know?

Turned out that my customer was only slightly behind the latest PHP version, at least that seemed to be the case at first sight.

No big deal, I'll update ...

A Powerless Administrator 🙄

Don't touch any WordPress setup before you have server access!

Sometimes, your customers will only provide you with an account as a WordPress administrator, which is fine unless there is any fatal error.

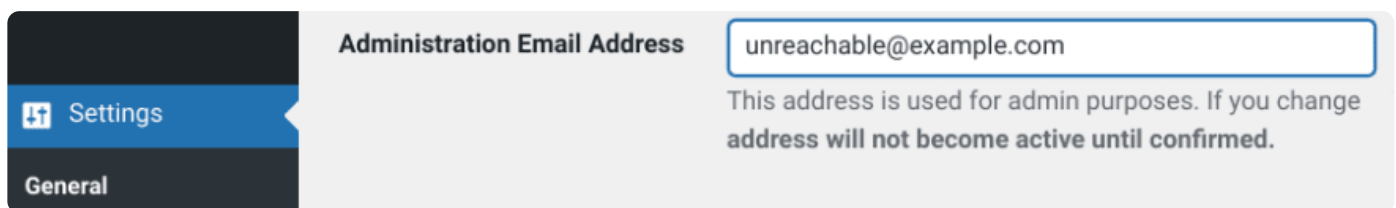
Housekeeping WordPress

- WordPress administration access
- webspace administration:
 - access the file system
 - view and edit PHP software version
 - database access
 - error logs

Recovery Email

In case of a fatal error, WordPress will send an email to the admin address, informing about the issue and including a secret link to activate [recovery mode](#).

Too bad if you're *an* administrator (a user having administrative rights) but not *the* administrator ("Administration Email Address" in general settings).



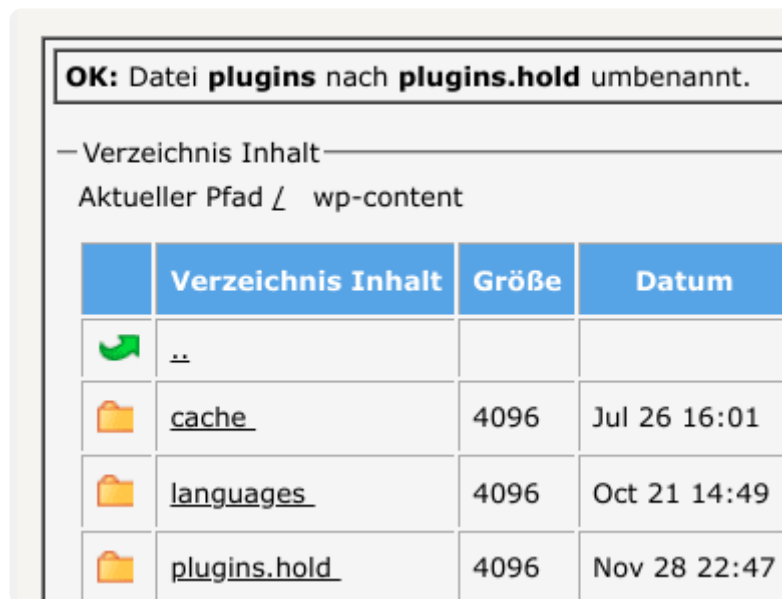
Too bad if you're set up an email address that no longer works or never did. Maybe a new spam prevention policy no longer allows PHP to send emails unless the domain matches your own domain, but you entered a gmail address.

Maybe you never cared about [WP Mail SMTP](#) because the built-in mailer used to work when you set up WordPress five years ago.

Whatever. The recovery email is the only way to get the recovery link. If you didn't get it, you can still try to use the database and the file system to recover your installation.

Alternative Recovery Strategy

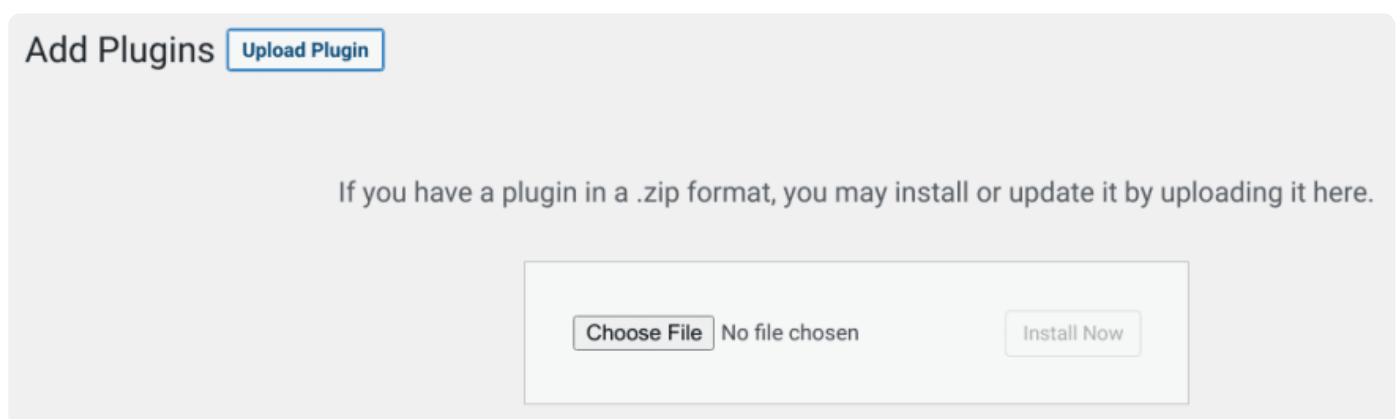
Like WordPress describes on their [troubleshooting page](#), you should disable all of your plugins, which can be done by renaming the plugins folder.



If you are able to log in again afterwards, you will see warning messages about missing plugins, and then you can re-enable them one by one until you see the problem. In my case I knew which update caused the problem, so I reactivated all but the troublesome plugin.

Reverting to an Older Version?

There doesn't seem to be a convenient way to install an older plugin version using the store, but you can download another version on the vendor's plugin page and install it by file upload.



While the older plugin version did not throw a fatal error, it didn't fully work either, so getting back to updating PHP at last.

Seemingly Minor PHP Update

I was mistaken to think that my customer was only slightly behind the latest PHP version. While 7.4 was the recommended stable release, the server still ran 7.3, so I updated to 7.4. made a coffee break. and loaded into WP-Admin to check again.

Nothing had changed.

Most web hosting providers offer the option to set different PHP versions for different domains and subdomains.

Sometimes you can even specify a specific PHP version **for any folder** in your file system! 💡

Nobody seemed to have noticed for some time, that this WordPress folder was stuck to PHP 7.1! No wonder things broke after the update.

Recovery

So after really updating to PHP 7.4, I first made another database backup just in case, and proceeded to update anything else. Updated WordPress core and all of the plugins.

Conclusion

Keep your system up to date!

If it's not your own system, make sure you (or another skilled person) will have access not only to WP-Admin, but also to the hosting provider's control panel, to the file system, and maybe have the phone number of their support hotline ready. ☎️

Back up your data before doing anything else!

Unless there is a legacy plugin that requires a legacy PHP version, proceed with the update:

Make sure to run the latest recommended stable PHP version, currently (in 2021) not 8.x yet (see [did I upgrade to PHP 8 too early?](#)), but no 5.x or any outdated 7.x version either.

PHP language level:	5.6 (variadic functions, argument unpacking)
	7.0 (return types, scalar type hints)
	7.1 (const visibility, nullables, multiple exceptions)
	7.2 (object type hint, abstract function override)
	7.3 (references in list assignments, flexible heredocs)
	7.4 (typed properties, short closures, null coalescing operator)
	8.0 (union types, named arguments, attributes, match expression)

After updating PHP, update WordPress core.

Only then update your themes and plugins.

When everything works, you can add further plugins that you might think you need, so you have more chances to fail once they release an update. 😊

Or you can finally start designing and blogging!

Discussion (4)



vladi160 • Dec 11 '21



Just turn on log errors and use FTP or hosting panel access to rename the plugin folder, if there is an error.

You can see which plugin in the error message .

Read plugin requirements.

Nothing complicated



Ingo Steinke 🌟 • Dec 11 '21



Yeah, I'll forward your message to my customer so they know how to deal with update notifications next time they open up wp-admin to write a post!

But seriously, renaming the plugin folder is the way to go if you have access to the file system. Otherwise, and if you're the owner and get the recovery mail, clicking on the recovery link will do a similar thing by deactivating all plugins.

As a regular WordPress user who doesn't know about FTP, PHP, etc. they probably wouldn't understand the requirements, and if they did, they'd have a hard time even finding out the current PHP version.

So "nothing complicated" is true from a developer's perspective.



vladi160 • Dec 11 '21



I will make (may be soon) a plugin just for that.

1. just open a specific URL - url dot com?id=UNIQUE_KEY; 2. PHP check for server errors -> if any, identify from where -> if it a plugin or theme, rename its folder



Khawaja_Awais • Dec 13 '21



✕ Instead of worrying about recovery and administrator email, The better way would be to just install easy update manager and disallow all updates so user does not mess anything up and you can schedule a monthly update for everything. It won't hurt to make an All in one migration backup or you can write a script that automatically does that. That is what I always do

[Code of Conduct](#) • [Report abuse](#)



Ingo Steinke

Web Development, Sustainability, Art and Music, Nature and Travel, Sustainability

LOCATION

Germany

WORK

Creative Web Developer at Ingo Steinke

JOINED

21 Sep 2019

More from [Ingo Steinke](#)

Animated Gradient Text Color

[#webdev](#) [#showdev](#) [#css](#) [#tutorial](#)

Aspect ratio: no need for container units!

[#css](#) [#html](#) [#webdev](#) [#todayilearned](#)

CSS :has(.parent-selectors)

[#css](#) [#webdev](#) [#todayilearned](#) [#javascript](#)