Considerations when running 5.4 on PHP7

Given that eZ Publish *(5.x series)* was created before PHP 7 came out, there are a few things to be aware of and consider when you want to run it on PHP 7.

> **Alternative: Running eZ Platform & eZ Platform Enterprise with Legacy Bridge**
> Did you know that you can also consider running eZ Publish legacy side by side with eZ Platform, and that this is supported together with eZ Publish Enterprise? It provides new features, improved security, simplified migration paths, fully supports PHP 7 *(including 7.2 and higher)*, and can be a good alternative to 5.4 which does not receive new features anymore.

> **Why PHP 7.2 and higher won't be officially supported on 5.4**
> PHP 7.x series has added more and more deprecations to cleanup the language for each new release. As 5.4 is a stable release, and one that has a legacy from way before PHP7 came to be, we can not risk BC breaks on 5.4 by attempting to fix these deprecations. However in the newer feature releases of eZ Publish legacy, support for this is progressively being added by both eZ and the community, making upgrading to eZ Platform 2.5LTS + legacy bridge 2.x + eZ Publish legacy 2019.03.x a viable alternative, *if* you are up for small adapting your own extensions for changes in PHP as well.
>
> So while we will try to make sure 5.4 is able to boot on higher version of PHP 7.x, it won't be officially supported for use 7.2 and higher in production. In practice this means if you hit issues which can be attributed to PHP version, and we can't fix it without breaking BC, then we will ask you to either upgrade eZ install or downgrade PHP.

Deprecation warnings

eZ Platform, including platform parts bundled with eZ Publish Platform 5.x, was made for more recent versions of PHP and tested for it early on. eZ Publish legacy, on the other hand, was first created back in 2001 and has a long range of extensions built for it closely depending on how it was written. And as eZ Publish is considered mature software, backwards compatibility breaks are avoided as much as possible. This means that there are a few deprecation warnings to be expected when running eZ Publish *(legacy)* on PHP 7 which won't be fixed as they risk breaking extensions. This includes:

- Class constructor names, example:

```
"Methods with the same name as their class will not be constructors in a future version of PHP;
eZContentObjectVersion has a deprecated constructor"
```

Importance of testing when upgrading to PHP7

PHP7 is a large change to the PHP language, and while it does provide a real world 2x - 2.6x improvement in performance, we can not stress enough the importance of testing your whole site/application when upgrading. You should essentially regard it as a re-launch.

Remember:

1. We try our best to cover all possible cases, especially in eZ Platform, our newest version. However, our QA and automated testing does not cover all the possible use case or hardware and software combinations you might be using eZ Publish with.
2. Extensions made by the community and shared on http://projects.ez.no/ or other places *(github/composer, ...)* might or might not have tested on PHP 7 yet.
3. Your own code base, including all your extensions, will need to be adapted for PHP 7.

So as a Partner make sure to stay on top of this when switching to PHP7. Make sure to gradually phase it in for development, then staging/acceptance/qa environment, before upgrading production environment. And if in doubt, we are here to help!