

YAPHPO Information

YAPHPO is "Yet Another PHP Obfuscator". Based originally on the code of "PHPCodeBuster" by Martijn Loots, YAPHPO's real strength is found in the framework of code developed around this core.

This framework of code provides much needed customisability for PHP developers that need to obfuscate their code. The key feature is the 'stepped obfuscation'.

There is nothing worse than feeding your code into an obfuscator and being given back a lump of obfuscated code that you can only hope is correct. Our stepped obfuscation allows you to ensure that the process worked correctly. For example, the first step may involve just obfuscating variable names. This way, you can compare your original code with the output from this step and not be overcome by differences. You can actually ensure that at this point, only the variable names have been changed (and changed correctly). The second step may involve removing comments. Now all you need to do is compare the output from the first step with the output from the second step and ensure that the comments and only the comments have been removed.

This is a great way to quickly get piece of mind that your code will still work after obfuscation! Of course, you don't have to go through all of that process if you don't want to. All of the 'steps' are done at once, but stored away in separate directories. If you like, just take the final directory and you have your completely obfuscated source.

Another important thing about obfuscating your code is that some variables and functions should not be obfuscated. For example, if your code is a module for a Content Management System of sorts, then you obviously don't want to change calls to functions that are outside of your control. YAPHPO deals with this by allowing you to define 'include' and 'exclude' lists for each different type of obfuscation. For example, you may use a library that provides all of your database functions and so you do not want to change calls to those particular functions. You can add each of those function names to the exclude list and they will not be affected by the obfuscation process. Better still, if they all start with 'db_', for example, just add 'db_*' to the list. Yes, it supports wildcards.

The reverse of this is also true: maybe your code provides an API for other developers. You can add your 'public' functions and variables to the exclude list and that way everyone else can still use these functions and variables that you provide, while not being able to read the rest of your code.

Unlike a lot of older obfuscators, YAPHPO uses PHP's own tokenizer. What does this mean? A lot of older obfuscators attempted to parse out your code and try and work out what each portion of your code was. YAPHPO, on the other hand, essentially passes your code block to PHP itself, which parses, analyses and marks up your code before passing it back to YAPHPO. No more trying to make sure that your code is acceptable to the obfuscator: if PHP understands your code, then you're all but guaranteed that YAPHPO will, too. The other great thing about this is that it's faster, too!

YAPHPO will work on an entire directory (plus subdirectories) of source code for you, so you don't even need to work on each file one at a time. The powerful 'include' and 'exclude' lists are usable for files, too, so that you can stop obfuscating and entire file, group of files or directories easily.

Unlike any other obfuscators that we know of, YAPHPO will even obfuscate your constants.

Finally, because YAPHPO outputs entirely valid PHP (though practically unreadable to a normal person) there is no need to make sure that there any particular extensions installed on the servers where your scripts will run. Other PHP encoders that are not obfuscators 'compile' the code but require extensions installed to let them run on the server. By using YAPHPO, you don't need to worry about these extensions being installed.

Best of all, YAPHPO is entirely free!

{mospagebreak title=Features}

What are the key features of YAPHPO that makes it ideal for your PHP obfuscation?

- It uses the PHP tokenizer itself: YAPHPO does not attempt to parse your code. Instead, your code gets passed to PHP itself which analyses it, marks it up then returns it for YAPHPO to do its work.
- Obfuscation of constants: a feature that we have not found in any of the other PHP obfuscators.
- The powerful 'include' and 'exclude' lists. Choose exactly which files, directories, functions, variables, classes and so on to obfuscate.
- Exclude those that you import from your CMS or database library, for example, as well as any functions that you wish to make available to other developers.
- Step-by-Step obfuscation: throughout the process you are easily able to ensure that it is working as expected. If you

like, you can run file comparisons between the results of each step of the obfuscation.

- Output is valid PHP: therefore, there is no requirement for any particular extensions to be installed on the server that your code is going to run on.
- It's Free and licensed under the GPL. Just to allay any fears: no, that does not mean that your program will have to be GPL, just because you use YAPHPO. Yes, you can use YAPHPO on your commercial products without any fear of being required to release the code to anyone.

{mospagebreak title=Links (including Downloads)}

Here are a number of links for you for YAPHPO.

- Download YAPHPO
- Issue tracking: YAPHPO is a public project in the AsIs Technologies Issue Tracker.
- Forum: if you need help running, using or extending YAPHPO, please visit the YAPHPO Forum.
- The original PHPCodeBuster project: the core of YAPHPO was based on this code.