



PHP4 to PHP 5 – the benefits of making the switch.

Table of Contents

Why php4?	1
Why php5?	2
Faster	2
Security	2
Object Enabled	2
Exceptions	3
Extensions	3
DOM	4
SPL	4
PDO	4
Incompatibilities	4
New Keywords	4
Mysql	5
Register Globals	5
Magic Quotes	5
Conclusion	6

Why php4?

Php4 was a huge success, largely in part to its ease of use for beginners, and advanced features for the power users. Php4 ended up powering a large majority of the web, from simple home pages to large enterprise applications.

The success of version 4, and its ability to be 'good enough', held a lot of projects back from moving onto the new and improved version 5.

Php4 has now reached its 'end of life' and will no longer be officially supported. This means any bugs that are left in the codebase may never be fixed, exploits that can be taken advantage of will not be closed, and there will be no improvements to the speed and efficiency of the engine.

This being the case, we advise that all code currently running on php4 be upgraded to run on php5.





Why php5?

Php5 is a complete rewrite of php's underlying operating engine, the Zend Engine 2.0. This rewrite brings with it many new features, which are discussed below.

Faster

The new engine is faster and more efficient at running the same code.

Security

The new engine is more secure by design. The notorious 'features' of php4 that would allow an attack vector have been removed, see the incompatibility section, as well as further enhancements to the engine for better security.

New features allow developers to make their applications more secure. The Filter extension (<http://php.net/filter>) allows a developer to validate and sanitise data against pre-set filters, such as an email address or a url.

```
$email = "not.an.email.address";

if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo ("'$email' is not a valid email address.");
}
```

Object Enabled

Php5 has had its entire object system rewritten from the ground up and is now much more of an 'object enabled' language¹. There are now many more OO features in the language that allow well known and used Object Orientated design methods to be used.

Objects now have a range of predetermined functions to work with objects:

```
__construct()
```

This function is called automatically when an object is created. This allows for code to be created to set the object up; for instance, a database object could use its construct function to create a

¹Object enabled, as opposed to an object orientated language, because PHP still allows non-object based code to be written.



connection to the database ready for any queries. In php4, the construct function had to share its name with the name of the object to be recognised and executed.

`__destruct()`

This function is called automatically when an object is destroyed. In the context of a database object, this could close any open connections and clean up any data left in memory.

`__get()`, `__set()`, `__call()`

These functions can be attached to an object and allow for virtual calls. If a call is made to a function that does not exist within an object, this will be passed to the `__call()` function so you can determine how these will be handled. Any attempts to declare or return a variable in an object can also be handled by `__set()` and `__get()` respectively.

Exceptions

Exceptions are a common design pattern in other languages, and now php5 has them. They can be used, as the name suggests, to deal with exceptions within the code - things that are not expected to happen, but may happen. This allows developers to provide an extra layer of functionality for unforeseen circumstances that may otherwise cause a crash, or worse, an exploit.

Extensions

Php5 comes with some new extensions which allow the language to stay relevant with the fast-paced nature of the internet.

SimpleXML

This extension provides a simple way of parsing and working with XML files.

```
$xml_string = <<<XML
<?xml version='1.0' standalone='yes'?>
<features>
<feature>
  <name>PHP 4</name>
  <description>Old release of PHP.</description>
</feature>
<feature>
  <name>PHP 5</name>
```



```
<description>New release of PHP.</description>
</feature>
</features>
XML;
```

```
$xml = new SimpleXMLElement($xml_string);

echo $xml->feature[1]->name; // Prints out "PHP 5"
```

DOM

The new DOM model allows the creation and modification of document object models which can then be translated into html or xml; which can be used to generate parts of a web-page, or to create the xml which would then later be used by the SimpleXML module.

SPL

The SPL (Standard PHP Library) has been extended to provide a range of base objects for dealing with data-structures, iterators, array objects, and file objects. These can all be extended with custom code to make use of the base features while still allowing complete customisation. Utilising the SPL can speed up development by not requiring developers to build generic functionality into their code.

PDO

This extension provides a data-access abstraction layer. This allows multiple backend database engines to be used with the same functions for sending and receiving data.

Incompatibilities

New Keywords

There are a few new reserved keywords in php5. As these are predetermined, these words cannot be used by developers in their code:





abstract
catch
clone
implements
interface
instanceof
namespace
private
protected
public
throw
try

__DIR__
__NAMESPACE__

Mysql

The traditional mysql module is now deprecated within php. Instead, the far more advanced mysqli module should be used, or the new PDO extension.

The old module is still available if it is needed, but not recommended.

Register Globals

Register globals has been deprecated in the new version of php. It can still be used, but by default it is off and is recommended not for use. As such, previous scripts that relied on native injections of global variables will no longer work.

Magic Quotes

Magic quotes, another relied upon feature of php4, has also been deprecated in php5. As such, all data must now be manually validated and sanitised at run time, instead of relying on this to occur behind the scenes.





Conclusion

There are a few other changes in the language, including the operation of a few functions which have changed, that have not been mentioned here. However, it can be seen, the features of php5 more than outweigh the effort needed to upgrade your scripts to work with the new engine. As most code that worked previously will still continue to work, only faster and more securely, there is very little reason to stay on php4.

