Deprecated features in PHP 7.2.x

- Unquoted strings
- png2wbmp and jpeg2wbmp
- INTL_IDNA_VARIANT_2003 variant
- __autoload method
- track_errors ini setting and $php_errormsg variable
- create_function function
- mbstring.func_overload ini setting
- (unset) cast
- parse_str without a second argument
- gmp_random function
- each function
- assert with a string argument
- $errcontext argument of error handlers
- read_exif_data function

Unquoted strings

Unquoted strings that are non-existent global constants are taken to be strings of themselves. This behaviour used to emit an `E_NOTICE`, but will now emit an `E_WARNING`. In the next major version of PHP, an `Error` exception will be thrown instead.

```
<?php


var_dump(NONEXISTENT);
```

```
/* Output:
Warning: Use of undefined constant NONEXISTENT - assumed 'NONEXISTENT' (this will throw an
Error in a future version of PHP) in %s on line %d
string(11) "NONEXISTENT"
*/
```

`png2wbmp` and `jpeg2wbmp`

The `png2wbmp` and `jpeg2wbmp` functions from the GD extension have now been deprecated and will be removed in the next major version of PHP.

`INTL_IDNA_VARIANT_2003` variant

The Intl extension has deprecated the `INTL_IDNA_VARIANT_2003` variant, which is currently being used as the default for `idn_to_ascii` and `idn_to_utf8`. PHP 7.4 will see these defaults changed to `INTL_IDNA_VARIANT_UTS46`, and the next major version of PHP will remove `INTL_IDNA_VARIANT_2003` altogether.

`__autoload` method

The `__autoload` method has been deprecated because it is inferior to `spl_autoload_register` (due to it not being able to chain autoloaders), and there is no interoperability between the two autoloading styles.

`track_errors` ini setting and `$php_errormsg` variable

When the `track_errors` ini setting is enabled, a `$php_errormsg` variable is created in the local scope when a non-fatal error occurs. Given that the preferred way of retrieving such error information is by using `error_get_last`, this feature has been deprecated.

`create_function` function

Given the security issues of this function (being a thin wrapper around `eval`), this dated function has now been deprecated. The preferred alternative is to use anonymous functions.

`mbstring.func_overload` ini setting

Given the interoperability problems of string-based functions being used in environments with this setting enabled, it has now been deprecated.

`(unset)` cast

Casting any expression to this type will always result in ENTITY, and so this superfluous casting type has now been deprecated.

`parse_str` without a second argument

Without the second argument to `parse_str`, the query string parameters would populate the local symbol table. Given the security implications of this, using `parse_str` without a second argument has now been deprecated. The function should always be used with two arguments, as the second argument causes the query string to be parsed into an array.

`gmp_random` function

This function generates a random number based upon a range that is calculated by an unexposed, platform-specific limb size. Because of this, the function has now been deprecated. The preferred way of generating a random number using the GMP extension is by `gmp_random_bits` and `gmp_random_range`.

`each` function

This function is far slower at iteration than a normal `foreach`, and causes implementation issues for some language changes. It has therefore been deprecated.

`assert` with a string argument

Using `assert` with a string argument required the string to be `eval`'ed. Given the potential for remote code execution, using `assert` with a string argument has now been deprecated in favour of using boolean expressions.

`$errcontext` argument of error handlers

The `$errcontext` argument contains all local variables of the error site. Given its rare usage, and the problems it causes with internal optimisations, it has now been deprecated. Instead, a debugger should be used to retrieve information on local variables at the error site.

`read_exif_data` function

The `read_exif_data` alias has been deprecated. The `exif_read_data` function should be used instead.

□□□ #2
□ □□□ □□□ 2026-10-31 17:33:56 UTC
□ □□□ □□□ 2026-10-31 17:33:57 UTC