

Focus search box

[List of other reserved words »](#)

[« Predefined Classes](#)

- [PHP Manual](#)
- [Appendices](#)
- [List of Reserved Words](#)

Change language: English ▼

[Submit a Pull Request](#) [Report a Bug](#)

Predefined Constants ¶

Core Predefined Constants ¶

These constants are defined by the PHP core. This includes PHP, the Zend engine, and SAPI modules.

PHP_VERSION (string)

The current PHP version as a string in "major.minor.release[extra]" notation.

PHP_MAJOR_VERSION (int)

The current PHP "major" version as an integer (e.g., int(5) from version "5.2.7-extra").

PHP_MINOR_VERSION (int)

The current PHP "minor" version as an integer (e.g., int(2) from version "5.2.7-extra").

PHP_RELEASE_VERSION (int)

The current PHP "release" version as an integer (e.g., int(7) from version "5.2.7-extra").

PHP_VERSION_ID (int)

The current PHP version as an integer, useful for version comparisons (e.g., int(50207) from version "5.2.7-extra").

PHP_EXTRA_VERSION (string)

The current PHP "extra" version as a string (e.g., '-extra' from version "5.2.7-extra"). Often used by distribution vendors to indicate a package version.

PHP_ZTS (int)

PHP_DEBUG (int)

PHP_MAXPATHLEN (int)

The maximum length of filenames (including path) supported by this build of PHP.

PHP_OS (string)

The operating system PHP was built for.

PHP_OS_FAMILY (string)

The operating system family PHP was built for. One of 'Windows', 'BSD', 'Darwin', 'Solaris', 'Linux' or 'Unknown'. Available as of PHP 7.2.0.

PHP_SAPI (string)

The Server API for this build of PHP. See also [php_sapi_name\(\)](#).

PHP_EOL (string)

The correct 'End Of Line' symbol for this platform.

PHP_INT_MAX (int)

The largest integer supported in this build of PHP. Usually int(2147483647) in 32 bit systems and int(9223372036854775807) in 64 bit systems.

PHP_INT_MIN (int)

The smallest integer supported in this build of PHP. Usually int(-2147483648) in 32 bit systems and int(-9223372036854775808) in 64 bit systems. Usually, PHP_INT_MIN === ~PHP_INT_MAX.

PHP_INT_SIZE (int)

The size of an integer in bytes in this build of PHP.

PHP_FLOAT_DIG (int)

Number of decimal digits that can be rounded into a float and back without precision loss. Available as of PHP 7.2.0.

PHP_FLOAT_EPSILON (float)

Smallest representable positive number x , so that $x + 1.0 \neq 1.0$. Available as of PHP 7.2.0.

PHP_FLOAT_MIN (float)

Smallest representable *positive* floating point number. If you need the smallest representable *negative* floating point number, use `- PHP_FLOAT_MAX`. Available as of PHP 7.2.0.

PHP_FLOAT_MAX (float)

Largest representable floating point number. Available as of PHP 7.2.0.

DEFAULT_INCLUDE_PATH (string)**PEAR_INSTALL_DIR** (string)**PEAR_EXTENSION_DIR** (string)**PHP_EXTENSION_DIR** (string)

The default directory where to look for dynamically loadable extensions (unless overridden by [extension_dir](#)). Defaults to `PHP_PREFIX` (or `PHP_PREFIX . "\\ext"` on Windows).

PHP_PREFIX (string)

The value `--prefix` was set to at configure. On Windows, it is the value `--with-prefix` was set to at configure.

PHP_BINDIR (string)

The value `--bindir` was set to at configure. On Windows, it is the value `--with-prefix` was set to at configure.

PHP_BINARY (string)

Specifies the PHP binary path during script execution.

PHP_MANDIR (string)

Specifies where the manpages were installed into.

PHP_LIBDIR (string)**PHP_DATADIR** (string)**PHP_SYSCONFDIR** (string)**PHP_LOCALSTATEDIR** (string)**PHP_CONFIG_FILE_PATH** (string)**PHP_CONFIG_FILE_SCAN_DIR** (string)**PHP_SHLIB_SUFFIX** (string)

The build-platform's shared library suffix, such as "so" (most Unixes) or "dll" (Windows).

PHP_FD_SETSIZE (string)

The maximum number of file descriptors for select system calls. Available as of PHP 7.1.0.

E_ERROR (int)

[Error reporting constant](#)

E_WARNING (int)

[Error reporting constant](#)

E_PARSE (int)

[Error reporting constant](#)

E_NOTICE (int)

[Error reporting constant](#)

E_CORE_ERROR (int)

[Error reporting constant](#)

E_CORE_WARNING (int)

[Error reporting constant](#)

E_COMPILE_ERROR (int)

[Error reporting constant](#)

E_COMPILE_WARNING (int)

[Error reporting constant](#)

E_USER_ERROR (int)

[Error reporting constant](#)

E_USER_WARNING (int)

[Error reporting constant](#)

E_USER_NOTICE (int)

[Error reporting constant](#)

E_RECOVERABLE_ERROR (int)

[Error reporting constant.](#)

E_DEPRECATED (int)

[Error reporting constant.](#)

E_USER_DEPRECATED (int)

[Error reporting constant.](#)

E_ALL (int)

[Error reporting constant](#)

E_STRICT (int)

[Error reporting constant](#)

__COMPILER_HALT_OFFSET__ (int)

true (bool)

See [Booleans](#).

false (bool)

See [Booleans](#).

null (null)

See [Null](#).

PHP_WINDOWS_EVENT_CTRL_C (int)

A Windows CTRL+C event. Available as of PHP 7.4.0 (Windows only).

PHP_WINDOWS_EVENT_CTRL_BREAK (int)

A Windows CTRL+BREAK event. Available as of PHP 7.4.0 (Windows only).

See also: [Magic constants](#).

Standard Predefined Constants ¶

All constants from [core extensions](#) are defined in PHP by default.

[+ add a note](#)

User Contributed Notes 1 note

[up](#)

[down](#)

-1

[rdcapasso ¶](#)

9 years ago

Volker's `getOS()` function needs to have the order of cases changed in the switch statement since "darwin" contains "win", which means that both "windows" and "darwin" will return `self::OS_WIN`. I've moved the 'dar' case above the 'win' case:

```
<?php
class System {

    const OS_UNKNOWN = 1;
    const OS_WIN = 2;
    const OS_LINUX = 3;
    const OS_OSX = 4;

    /**
     * @return int
     */
    static public function getOS() {
        switch (true) {
            case strpos(PHP_OS, 'DAR'): return self::OS_OSX;
            case strpos(PHP_OS, 'WIN'): return self::OS_WIN;
            case strpos(PHP_OS, 'LINUX'): return self::OS_LINUX;
            default : return self::OS_UNKNOWN;
        }
    }
}
```