Other Changes

- Core changes
  - Core
  - CGI/CLI
  - PDO_ODBC
- Changes in SAPI Modules
  - CLI
  - FPM
- Changed Functions
  - Intl
  - LDAP
  - libxml
  - OpenSSL
  - PCNTL
  - PDO_PGSQL
  - PostgreSQL
  - POSIX
  - Reflection
  - Sockets
  - Zlib
- Other Changes to Extensions
  - cURL
  - Fileinfo
  - Intl
  - Lexbor
  - PCRE
  - PDO_Sqlite
  - Readline
  - Reflection

- Changes to INI File Handling
  - Core
  - Opcache
  - OpenSSL
- Performance
  - Core
  - Intl
  - MBString
  - Opcache
  - Reflection
  - SPL
  - Standard
  - XML

Core changes

Core

The high resolution timer (`hrtime`) on macOS now uses the recommended `clock_gettime_nsec_np(CLOCK_UPTIME_RAW)` API instead of `mach_absolute_time()`.

CGI/CLI

The `-z` or `--zend-extension` option has been removed as it was non-functional. Use `-d zend_extension=[path]` instead.

PDO_ODBC

The fetch behaviour for larger columns has been changed. Rather than fetching 256 byte blocks, PDO_ODBC will try to fetch a larger block size; currently, this is the page size minus string overhead. Drivers that return SQL_NO_TOTAL in SQLGetData are also better handled as well. This should improve compatibility and performance.

Changes in SAPI Modules

CLI

Trying to set a process title that is too long with `cli_set_process_title` will now fail instead of silently truncating the given title.

Added a new `--ini=diff` option to print INI settings changed from the builtin default.

FPM

FPM with httpd ProxyPass optionally decodes the full script path. Added fastcgi.script_path_encoded INI setting to prevent this new behavior.

FPM access log limit now respects log_limit value.

Changed Functions

Intl

`grapheme_extract` properly assigns `next` value when skipping over invalid starting bytes. Previously there were cases where it would point to the start of the grapheme boundary instead of the end.

`transliterator_get_error_code`, `transliterator_get_error_message`, TransLiterator::getErrorCode, and TransLiterator::getErrorMessage have dropped ENTITY from the return type union. Returning ENTITY was actually never possible.

The following functions now support a `locale` argument: `grapheme_strpos`, `grapheme_stripos`, `grapheme_strrpos`, `grapheme_strripos`, `grapheme_substr`, `grapheme_strstr` and `grapheme_stristr`

LDAP

`ldap_get_option` now accepts a ENTITY connection, like `ldap_set_option`, to allow retrieval of global options.

libxml

`libxml_set_external_entity_loader` now has a formal return type of `true`.

OpenSSL

`openssl_public_encrypt` and `openssl_private_decrypt` have a new parameter `digest_algo` that allows specifying the hash digest algorithm for OAEP padding.

`openssl_sign` and `openssl_verify` have a new parameter `padding` to allow using more secure RSA PSS padding.

`openssl_cms_encrypt` `cipher_algo` parameter can be a string with the cipher name. That allows to use more algorithms including AES GCM cipher algorithms for auth enveloped data.

PCNTL

`pcntl_exec` now has a formal return type of `false`.

`pcntl_waitid` takes an additional resource_usage argument to gather various platform specific metrics about the child process.

PDO_PGSQL

Pdo\Pgsql::copyFromArray now supports `iterable` inputs.

Pdo\Pgsql::setAttribute and Pdo\Pgsql::prepare support setting `PDO::ATTR_PREFETCH` to 0 which enters lazy fetch mode. In this mode, statements cannot be run in parallel.

PostgreSQL

`pg_copy_from` now supports `iterable` inputs.

`pg_connect` checks if the connection_string argument contains any null byte.

`pg_close_stmt` checks if the statement_name argument contains any null byte.

POSIX

`posix_ttyname` sets last_error to EBADF when encountering an invalid file descriptor.

`posix_isatty` raises an `E_WARNING` message when encountering an invalid file descriptor.

`posix_fpathconf` checks invalid file descriptors and sets last_error to EBADF and raises an `E_WARNING` message.

Reflection

The output of ReflectionClass::__toString for enums has changed to better indicate that the class is an enum, and that the enum cases are enum cases rather than normal class constants.

The output of ReflectionProperty::__toString for properties with hooks has changed to indicate what hooks the property has, whether those hooks are final, and whether the property is virtual. This also affects the output of ReflectionClass::__toString when a class contains hooked properties.

Sockets

`socket_create`/`socket_bind` can create `AF_PACKET` family sockets.

`socket_getsockname` gets the interface index and its string representation with `AF_PACKET` socket.

Zlib

The `use_include_path` argument for the `gzfile`, `gzopen` and `readgzfile` functions has been changed from `int` to `bool`.

`gzfile`, `gzopen` and `readgzfile` functions now respect the default stream context.

Other Changes to Extensions

cURL

`curl_setopt` with `CURLOPT_FOLLOWLOCATION` option's value no longer is treated as boolean but integer to handle `CURLFOLLOW_OBEYCODE` and `CURLFOLLOW_FIRSTONLY`.

Fileinfo

Upgraded file from 5.45 to 5.46.

The return type of `finfo_close` has been changed to `true`, rather than `bool`.

Intl

Intl's internal error mechanism has been modernized so that it indicates more accurately which call site caused what error. Moreover, some ext/date exceptions have been wrapped inside a `IntlException` now.

Lexbor

An always enabled lexbor extension is added. It contains the lexbor library that was separated from ext/dom for being reused among other extensions. The new extension is not directly exposed to userland.

PCRE

Upgraded pcre2lib from 10.44 to 10.46.

PDO_Sqlite

Increased minimum release version support from 3.7.7 to 3.7.17.

Readline

The return types of `readline_add_history`, `readline_clear_history`, and `readline_callback_handler_install` have been changed to `true`, rather than `bool`.

Reflection

`ReflectionConstant` is no longer final.

Changes to INI File Handling

Core

Added fatal_error_backtraces to control whether fatal errors should include a backtrace.

Added startup-only max_memory_limit INI setting to control the maximum memory_limit that may be configured at startup or runtime. Exceeding this value emits a warning, unless set to -1, and sets memory_limit to the current max_memory_limit instead.

Opcache

Added opcache.file_cache_read_only to support a read-only opcache.file_cache directory, for use with read-only file systems (e.g. read-only Docker containers). Best used with `opcache.validate_timestamps=0`, `opcache.enable_file_override=1`, and `opcache.file_cache_consistency_checks=0`.

> [!NOTE] A cache generated with a different build of PHP, a different file path, or different settings (including which extensions are loaded), may be ignored.

The default value of opcache.jit_hot_loop is now 61 (a prime) to prevent it from being a multiple of loop iteration counts. It is recommended that this parameter is set to a prime number.

Changing opcache.memory_consumption when OPcache SHM is already set up will now correctly report a failure instead of silently doing nothing and showing misleading values in PHPInfo.

OpenSSL

Added openssl.libctx to select the OpenSSL library context type. Either custom libctx for each thread can be used or a single global (default) libctx is used.

Performance

Core

Remove OPcodes for identity comparisons against booleans, particularly for the `match(true)` pattern.

Add OPcode specialization for `=== []` and `!== []` comparisons.

Creating exception objects is now much faster.

The parts of the code that used SSE2 have been adapted to use SIMD with ARM NEON as well.

Introduced the TAILCALL VM, enabled by default when compiling with Clang>=19 on x86_64 or aarch64. The TAILCALL VM is as fast as the HYBRID VM used when compiling with GCC. This makes PHP binaries built with Clang>=19 as fast as binaries built with GCC. The performance of the CALL VM, used with other compilers, has also improved considerably.

Intl

Now avoids creating extra string copies when converting strings for use in the collator.

MBString

The parts of the code that used SSE2 have been adapted to use SIMD with ARM NEON as well.

Opcache

Improved performance of fetching TLS variables in JIT'ed code in non-Glibc builds.

Reflection

Improved performance of the following methods: ReflectionProperty::getValue, ReflectionProperty::getRawValue, ReflectionProperty::isInitialized, ReflectionProperty::isInitialized, ReflectionProperty::setValue, ReflectionProperty::setRawValue

SPL

Improved performance of dimension accessors and methods of `SplFixedArray`.

Standard

Improved performance of array functions with callbacks (`array_find`, `array_filter`, `array_map`, `usort`, ...).

Improved performance of `urlencode` and `rawurlencode`.

Improved `unpack` performance with nameless repetitions by avoiding creating temporary strings and reparsing them.

Improved `pack` performance.

Minor improvements in `array_chunk` performance.

XML

Improved `XMLReader` property access performance.

Improved `XMLWriter` performance and reduced memory consumption.

□□□ #2
□ □□□ □□□ 2026-10-31 17:36:55 UTC
□ □□□ □□□ 2026-10-31 17:36:56 UTC