PHP 8.5: New Features from the Source

Daniel Scherzer

Daniel Scherzer

- Full-stack developer and open source contributor
- https://scherzer.dev
- http://github.com/DanielEScherzer
- PHP core contributor
  - ext/reflection maintainer
  - 8.5 release manager

Overview

- New features (not exhaustive)
  - Tooling that understands/validates the usage
  - Polyfills or workarounds
- Removals
- Deprecations and warnings (not exhaustive)
  - Not removals
  - Tooling that identifies/mitigates issues
- Questions

Language: more constant expressions (1)

```
<?php
#[\Attribute]
class Tests {
        public function __construct(
                private \Closure $cb
        ) {}
        public function run() { ($this->cb)(); }
}
#[Tests(static function () {
        assert(isPrime(2));
        assert(!isPrime(100));
})]
function isPrime( int $val = (int)'123' ): bool
        { /* Exercise for the reader */ return false; }

new \ReflectionFunction('isPrime')->getAttributes()[0]
        ->newInstance()->run();
```

- Attributes parameters
- Parameter and property defaults
- Global constants, class constants
- Access to private members
- Closures, casts, and first class callables

Language: more constant expressions (2)

```php
<?php
class StringUtils {
        public const LENGTH = strlen(...);
        public function __construct( public readonly string $str ) {}
        public function transformed(Closure $transform = strrev(...)) {
                return new StringUtils( $transform( $this->str ) );
        }

}

var_dump( (StringUtils::LENGTH)('example') );
$str = new StringUtils( 'ConFoo' );
var_dump( $str->transformed() );
var_dump( $str->transformed( strtoupper(...) ) );
```

PHPStan:

- Closures (2.1.32, ab704f4)
- FCC (2.1.32, f436abf)
- Casts (2.1.32, df5ede5)

Language: clone() $with

```php
<?php
class Demo {
        public function __construct(
                public public(set) readonly int $val
        ) {}
}

$d1 = new Demo( 5 );
$d2 = clone( $d1, [ 'val' => 42 ] );
var_dump( $d2 );
```

- __clone() called *before* $with applied
- Assignment in array order
- Visibility rules enforced
- Property hooks applied
- Dynamic properties validated, __set() called
- Lazy objects are made non-lazy
- Readonly unlocked:
  - Write in both __clone() and $with
  - Limit to once in each

PHPStan understanding: 2.1.32, 5a9471a

Language: pipe operator |>

```php
<?php
$caps = new class {
        public function __invoke( string $val ) {
                return strtoupper( $val );
        }
};
$result = [ '=w2bvNGI', 'lJXYgMXZ', 'wlGU' ]
        |> (static fn (array $a): string => implode('', $a))
        |> strrev(...)
        |> 'base64_decode'
        |> $caps;
var_dump( $result ); // "PIPES ARE COOL"
```

- Compile-time optimized away
- Any callable (that works with 1 parameter)

Tooling support

- NestedToPipeOperatorRector (2.2.6, 09ab43b)
    - Renamed in (2.2.7, 758adc7) to SequentialAssignmentsToPipeOperatorRector
- NestedFuncCallsToPipeOperatorRector (2.2.7, 8376d26)
- PHPStan understanding (2.1.32, e9df52e)
- Phan understanding (6.0.0, 0048e8b)

Language: attributes on constants

```php
<?php
namespace Some\Library {
        const VERSION = '1.2.3';
}

namespace {
        #[\Deprecated("Use Some\Library\VERSION")]
        const SOME_LIBRARY_VERSION =
                \Some\Library\VERSION;
}
```

- New: Attribute::TARGET_CONSTANT
- Updated: Attribute::TARGET_ALL
- New: ReflectionConstant::getAttributes()
- Only for compile-time constants

Tooling support

- PHPStan attribute validation (2.1.32, 6958f86)
- Rector (2.1.5, c5876ad)
  DeprecatedAnnotationToDeprecatedAttributeRector

Blog post with the history and implementation

Language: other improvements

Asymmetric visibility for **static** properties

```
<?php
class Example
        public private(set) static string $prop = 'Foo';
        protected(set) static int $otherThing = 0;
}
```

PHPStan 2.1.32 (8b2b97f):

- assign.staticPropertyPrivateSet
- assign.staticPropertyProtectedSet
- property.staticAsymmetricVisibility

Constructor property promotion for **final** properties

```
<?php
class A {
        public function __construct(
                final $prop
        ) {}
}
```

PHPStan 2.1.32 (e5de6b8):

- property.invalidPromoted (version detection)

New extension: ext/uri

- Scheme
- Username
- Password
- Host
- Port
- Path
- Query
- Fragment

- \Uri\Rfc3986\Uri
  - Formal grammar, invalid inputs rejected
  - Uniform Resource Identifier
  - Fixed specification
- \Uri\WhatWg\Url
  - Algorithmic, handling of malformed inputs
  - Uniform Resource Locators
  - Living document
- The RFC has details
- `league/uri-polyfill` package

Extension change: OPCache enabled

- --enable-opcache
- --disable-opcache
- zend_extension=opcache.so
- zend_extension=php_opcache.dll
- opcache_*() functions

- opcache.enable
- opcache.enable_cli

New INI: fatal_error_backtraces

- Respects #[\SensitiveParameter]
- Respects zend.exception_ignore_args
- Defaults to ON :)

- Class redeclaration
- Memory limit exhaustion
- Maximum execution time exhaustion

New Attribute: #[\NoDiscard]

```php
<?php
#[\Attribute(
        \Attribute::TARGET_METHOD
        |\Attribute::TARGET_FUNCTION
)]
final class NoDiscard {
        public function __construct(
                public readonly ?string $message = null
        ) {}
}
```

Language change: `(void)` casts

- Symfony 8.5 polyfills: 1.32.0, f3d7f32

Tooling support

- PHPStan: (2.1.28, f8ad453)
  - callable.resultDiscarded
  - method.resultDiscarded
  - staticMethod.resultDiscarded
- Phan:
  - PhanPluginUseReturnValueKnown (5.5.2, 8c414bd)
  - PhanNoDiscardReturnValueIgnored (6.0.0, 0048e8b)

New Attribute: #[\DelayedTargetValidation]

```php
<?php
#[\Attribute(\Attribute::TARGET_ALL)]
final class DelayedTargetValidation {
	public function __construct() {}
}
```

- Symfony 8.5 polyfills: PENDING, 17524a5
- Detailed explanation on my blog

Improved Attribute: #[\Override] properties

```php
<?php
abstract class Base {
        abstract public mixed $prop { get; }
}

class Child extends Base {
        #[\Override]
        public mixed $prop;
}

class Subchild extends Child {
        #[\Override]
        public mixed $property;
}
```

- PHPStan 2.1.32 (8b85e62)
  - Validation: property.override
  - Suggestion: property.missingOverride
  - Version: property.overrideAttribute
- Phan 6.0.0 (0048e8b)
- Class exists, cannot polyfill expansion of targets
  - #[\DelayedTargetValidation] use case

Improved Attribute: #[\Deprecated] traits

```php
<?php
#[\Deprecated]
trait DemoTrait {}

class DemoClass {
    use DemoTrait;
}
```

- Phan: PhanDeprecatedTrait (6.0.0, a4d67b6)
- PHPStan 2.1.32 (f0675a7)
  - target validation for classes
  - trait.deprecatedAttribute for versions
- Class exists, cannot polyfill expansion of targets
  - #[\DelayedTargetValidation] use case

New feature: persistent cURL handles

- curl_share_init_persistent()
- CurlSharePersistentHandle
- Share options fixed
  - No CURL_LOCK_DATA_COOKIE

- Not polyfillable
- Persist between requests

New feature: CHIPS

“Cookies Having Independent Partitioned State”

- setcookie() [array overload]
- setrawcookie() [array overload]
- session_set_cookie_params()
- session_get_cookie_params()
  - All with “partitioned”
- session_start()
  - “cookie_partitioned”

- Manual header()
- symfony/http-foundation (6.4.0, f13a4b1)

Added: FILTER_THROW_ON_FAILURE

- Alternative to FILTER_NULL_ON_FAILURE
- Throws \Filter\FilterFailedException
  - Extends \Filter\FilterException

- Prompted by feature request on GitHub: #16710
- Out-of-band signal for failures

Tooling support

- PHPStan inference (2.1.32, 193d801)
- PHPStan flag validation (2.1.39, 296dfee)

New functions: get_*_handler()

```php
<?php
function get_error_handler(): ?callable {
        $handler = set_error_handler(null);
        restore_error_handler();
        return $handler;
}

function get_exception_handler(): ?callable {
        $handler = set_exception_handler(null);
        restore_exception_handler();
        return $handler;
}
```

- Null when no handler set
- Symfony 8.5 polyfills: 1.32.0, d791faa

New functions: array_first() / array_last()

```php
<?php
function array_first(array $array): mixed {
        return $array === []
                ? null
                : $array[array_key_first($array)];
}

function array_last(array $array): mixed {
        return $array === []
                ? null
                : $array[array_key_last($array)];
}
```

- Null on empty arrays
- Symfony 8.5 polyfills: 1.33.0, fc474ff
- polyfills/array-first-array-last: 1.0.0, c2a004e

Migration: ArrayFirstLastRector (2.0.17, 4c3f175)

Skip nested writes: 2.2.9, b5e9d7b

Avoid unneeded casts: 2.3.0, 3288dc1

Skip unset: 2.3.0, e20f1c9

Skip top level writes: 2.3.2, 732278e

Removed: INI disable_classes

- Breaks engine expectations
- All core classes are safe

- SlevomatCodingStandard.PHP.ForbiddenClasses (6.3.0, 915a7cd)
- PHPCompatibility.IniDirectives.RemovedIniDirectives (10.0.0-alpha2, e3a71a2)

Warnings: Footguns for types

```
<?php
$nan = fdiv( 0, 0 );
var_dump( (string)$nan );

$aboveMax = PHP_INT_MAX + 1;
var_dump( (int)$aboveMax );

[$first, $second] = 123;
var_dump( $first, $second );
```

- NAN coercions:
  - settype()
  - Parameters and return values
  - (string), (bool), (array)
- Out of range float cast to int
  - (int) for NaN
  - PHP_INT_MAX + 1
- Destructuring non-arrays
  - Objects already though `Error`s
  - Scalar/null/resource just assigned null

Deprecated: confusing chr() and ord()

chr() outside of [0, 255]

Values are bitwise-AND with 255 (0xff) and that ASCII char is returned

Rector: ChrArgModuloRector (2.1.5, ef73f11)

PHPStan: argument.type updated (2.1.39, d07af42)

ord() with bad lengths

First character from a C-string is returned as an int

- Empty string -> NULL -> ‘’
- Multiple chars -> first

Rector: OrdSingleByteRector (2.1.7, 5ef04eb)

Handle missing arg: 2.2.10, 9f4509d

Deprecated: in-band Reflection failures

ReflectionClass::getConstant() when missing

- `false` indicates missing constant
- `false` is a valid constant value
- Guard with `::hasConstant()`

ReflectionProperty::getDefaultValue() when missing

- `null` indicates missing default
- `null` is a valid default
- Guard with `::hasDefaultValue()`

Deprecated: `$http_response_header`

Compile-time error

```php
<?php
$GLOBALS['http_response_header'] = 0;
var_dump( $http_response_header );
```

```php
<?php
$url = 'https://www.php.net/release-candidates.php?format=json';
$content = file_get_contents($url);
if (function_exists('http_get_last_response_headers')) {
        $http_response_header = http_get_last_response_headers();
}
// Safe to use $http_response_header
```

Deprecated: semicolon after `case`

```php
<?php
switch (substr(PHP_VERSION, 0, 3)) {
    case '8.5';
        echo "Deprecated\n";
        break;
    default;
        echo "Semicolon works\n";
        break;
}
```

- PHP CS Fixer: switch_case_semicolon_to_colon (1.12.0, 0e56077)
- PHPCS: 1.4.0 (baec904) PSR2.ControlStructures.SwitchDeclaration.WrongOpenercase
    - Message: 2.3.2 (931842a)
    - Autofix: 3.13.3 (d9766f8)
- PHPStan: backtick.deprecated (2.1.32, 9405233)
    - Autofix: 2.1.32 (89c7bf9)
- Rector: ColonAfterSwitchCaseRector (2.1.5, eda7704)

Deprecated: ArrayObject for objects

- Also ArrayIterator
- Break engine assumptions

- Phan: PhanDeprecatedConstructorObjectParamInternal (5.5.2, 76c8d10)

Deprecated: non-standard cast names

Replace:	With:
(binary)	(string)
(boolean)	(bool)
(double)	(float)
(integer)	(int)

- PHP CS Fixer: short_scalar_cast (1.12.0, 0e56077)
  - (binary) since 2.12.0 (5ac307d)
- PHPCS: SlevomatCodingStandard.PHP.TypeCast (4.5.0, 563b080)
  - Forbids (binary)
- PHPStan: cast.deprecated (2.1.32, d0a77d1)
- Rector: RenameCastRector (2.1.4, 33b12c2)

Deprecated: backticks for shell_exec()

```
<?php
$version = `php -v`;
echo $version;
```

- PHP CS Fixer: backtick_to_shell_exec (2.10.0, 45e7c85)
- PHPCS: Generic.PHP.BacktickOperator (2.7.0, 4543ab2)
- PHPStan strict rules: backtick.notAllowed (0.12.0, b7ae6e0)
    - PHPStan: backtick.deprecated (2.1.32, 9405233)
- Rector: ShellExecFunctionCallOverBackticksRector (2.2.2, ade4c81)

Deprecated: `null` as array offsets

- Also array_key_exists()
- Just use an empty string

- Phan: PhanTypeMismatchArgumentInternalProbablyReal (Reflection - PHP 8.0?)
  - PhanTypeInvalidDimOffset (5.3.0, 55745e6)
- Psalm: InvalidArrayAccess (0.1, 8c5adf4)
  - InvalidArrayOffset (0.2.66, 6c9cdd8)
  - NullArrayOffset / PossiblyNullArrayOffset (0.2.66, 1f57881)
- Rector: ArrayKeyExistsNullToEmptyStringRector (2.1.5, aeb8a67)

Deprecated: __debugInfo() with `null`

```
<?php
class Example {

        public function __debugInfo() {
                return null;
        }

}
```

Equivalent to an empty array

- Phan: PhanTypeMismatchReturn (2.0.0, b75c692)
  - PhanTypeMismatchReturnProbablyReal (3.1.0, 2279b77)
- Rector: NullDebugInfoReturnRector (2.1.3, 3dd70ad)

Deprecated: no-op closing functions

- curl_close()
- curl_share_close()
- finfo_close()
- imagedestroy()
- xml_parser_free()

- PHPStan 2.1.23:
  - curl_close() (fbb9cb6)
  - curl_share_close() (fbb9cb6)
  - finfo_close() (f43f8ea)
  - imagedestroy() (0c62fa9)
  - xml_parser_free() (fbb9cb6)
- Phan: PhanDeprecatedFunctionInternal, (Reflection - PHP 8.5)
- Rector: RemoveFuncCallRector (2.2.4, 4b30bc5)
  - WrapFuncCallWithPhpVersionIdCheckerRector (2.1.5, 66ac8c6)

Deprecated: Reflection*::setAccessible()

- ReflectionMethod::setAccessible()
- ReflectionProperty::setAccessible()

Since PHP 8.1, all methods/properties are accessible

- PHPStan stub data (2.1.23, fbb9cb6)
- Phan: PhanDeprecatedFunctionInternal, (Reflection - PHP 8.5)
- Rector: RemoveReflectionSetAccessibleCallsRector (2.1.3, 6d046c1)
  - Added to 8.1 config set

Deprecated: `null` directory arguments

- readdir()
- rewinddir()
- closedir()

`null` results in using the most recently opened directory stream, just pass that directly

- Phan: PhanTypeMismatchArgumentInternalProbablyReal (Reflection - PHP 8.0?)

Summary

PHP 8.5

- …has a lot of new language features:
  - Constant expressions can use closures, casts, and first class callables
  - Clone with for updating fields
  - Pipe operator for chaining callables
- …has significant attribute improvements:
  - Attributes on constants
  - NoDiscard attribute
- …comes with opcache and uri always available
- …includes a number of deprecations, though mitigation is simple

Thank You

Questions?