# CHANGELOG

## 7.2

* Deprecate parsing duplicate mapping keys whose value is null

## 7.1

* Add support for getting all the enum cases with !php/enum Foo

## 7.0

* Remove the !php/const: tag, use !php/const instead (without the colon)

## 6.3

* Add support to dump int keys as strings by using the Yaml::DUMP\_NUMERIC\_KEY\_AS\_STRING flag

## 6.2

* Add support for !php/enum and !php/enum \*->value
* Deprecate the !php/const: tag in key which will be replaced by the !php/const tag (without the colon) since 3.4

## 6.1

* In cases where it will likely improve readability, strings containing single quotes will be double-quoted

## 5.4

* Add new lint:yaml dirname --exclude=/dirname/foo.yaml --exclude=/dirname/bar.yaml option to exclude one or more specific files from multiple file list
* Allow negatable for the parse tags option with --no-parse-tags

## 5.3

* Added github format support & autodetection to render errors as annotations when running the YAML linter command in a Github Action environment.

## 5.1.0

* Added support for parsing numbers prefixed with 0o as octal numbers.
* Deprecated support for parsing numbers starting with 0 as octal numbers. They will be parsed as strings as of Symfony 6.0. Prefix numbers with 0o so that they are parsed as octal numbers.

Before:

Yaml::parse('072');

After:

Yaml::parse('0o72');

* Added yaml-lint binary.
* Deprecated using the !php/object and !php/const tags without a value.

## 5.0.0

* Removed support for mappings inside multi-line strings.
* removed support for implicit STDIN usage in the lint:yaml command, use lint:yaml - (append a dash) instead to make it explicit.

## 4.4.0

* Added support for parsing the inline notation spanning multiple lines.
* Added support to dump null as ~ by using the Yaml::DUMP\_NULL\_AS\_TILDE flag.
* deprecated accepting STDIN implicitly when using the lint:yaml command, use lint:yaml - (append a dash) instead to make it explicit.

## 4.3.0

* Using a mapping inside a multi-line string is deprecated and will throw a ParseException in 5.0.

## 4.2.0

* added support for multiple files or directories in LintCommand

## 4.0.0

* The behavior of the non-specific tag ! is changed and now forces non-evaluating your values.
* complex mappings will throw a ParseException
* support for the comma as a group separator for floats has been dropped, use the underscore instead
* support for the !!php/object tag has been dropped, use the !php/object tag instead
* duplicate mapping keys throw a ParseException
* non-string mapping keys throw a ParseException, use the Yaml::PARSE\_KEYS\_AS\_STRINGS flag to cast them to strings
* % at the beginning of an unquoted string throw a ParseException
* mappings with a colon (:) that is not followed by a whitespace throw a ParseException
* the Dumper::setIndentation() method has been removed
* being able to pass boolean options to the Yaml::parse(), Yaml::dump(), Parser::parse(), and Dumper::dump() methods to configure the behavior of the parser and dumper is no longer supported, pass bitmask flags instead
* the constructor arguments of the Parser class have been removed
* the Inline class is internal and no longer part of the BC promise
* removed support for the !str tag, use the !!str tag instead
* added support for tagged scalars.

Yaml::parse('!foo bar', Yaml::PARSE\_CUSTOM\_TAGS);

// returns TaggedValue('foo', 'bar');

## 3.4.0

* added support for parsing YAML files using the Yaml::parseFile() or Parser::parseFile() method
* the Dumper, Parser, and Yaml classes are marked as final
* Deprecated the !php/object: tag which will be replaced by the !php/object tag (without the colon) in 4.0.
* Deprecated the !php/const: tag which will be replaced by the !php/const tag (without the colon) in 4.0.
* Support for the !str tag is deprecated, use the !!str tag instead.
* Deprecated using the non-specific tag ! as its behavior will change in 4.0. It will force non-evaluating your values in 4.0. Use plain integers or !!float instead.

## 3.3.0

* Starting an unquoted string with a question mark followed by a space is deprecated and will throw a ParseException in Symfony 4.0.
* Deprecated support for implicitly parsing non-string mapping keys as strings. Mapping keys that are no strings will lead to a ParseException in Symfony 4.0. Use quotes to opt-in for keys to be parsed as strings.

Before:

$yaml = <<<YAML

null: null key

true: boolean true

2.0: float key

YAML;

Yaml::parse($yaml);

After:

$yaml = <<<YAML

"null": null key

"true": boolean true

"2.0": float key

YAML;

Yaml::parse($yaml);

* Omitted mapping values will be parsed as null.
* Omitting the key of a mapping is deprecated and will throw a ParseException in Symfony 4.0.
* Added support for dumping empty PHP arrays as YAML sequences:

Yaml::dump([], 0, 0, Yaml::DUMP\_EMPTY\_ARRAY\_AS\_SEQUENCE);

## 3.2.0

* Mappings with a colon (:) that is not followed by a whitespace are deprecated when the mapping key is not quoted and will lead to a ParseException in Symfony 4.0 (e.g. foo:bar must be foo: bar).
* Added support for parsing PHP constants:

Yaml::parse('!php/const:PHP\_INT\_MAX', Yaml::PARSE\_CONSTANT);

* Support for silently ignoring duplicate mapping keys in YAML has been deprecated and will lead to a ParseException in Symfony 4.0.

## 3.1.0

* Added support to dump stdClass and ArrayAccess objects as YAML mappings through the Yaml::DUMP\_OBJECT\_AS\_MAP flag.
* Strings that are not UTF-8 encoded will be dumped as base64 encoded binary data.
* Added support for dumping multi line strings as literal blocks.
* Added support for parsing base64 encoded binary data when they are tagged with the !!binary tag.
* Added support for parsing timestamps as \DateTime objects:

Yaml::parse('2001-12-15 21:59:43.10 -5', Yaml::PARSE\_DATETIME);

* \DateTime and \DateTimeImmutable objects are dumped as YAML timestamps.
* Deprecated usage of % at the beginning of an unquoted string.
* Added support for customizing the YAML parser behavior through an optional bit field:

Yaml::parse('{ "foo": "bar", "fiz": "cat" }', Yaml::PARSE\_EXCEPTION\_ON\_INVALID\_TYPE | Yaml::PARSE\_OBJECT | Yaml::PARSE\_OBJECT\_FOR\_MAP);

* Added support for customizing the dumped YAML string through an optional bit field:

Yaml::dump(['foo' => new A(), 'bar' => 1], 0, 0, Yaml::DUMP\_EXCEPTION\_ON\_INVALID\_TYPE | Yaml::DUMP\_OBJECT);

## 3.0.0

* Yaml::parse() now throws an exception when a blackslash is not escaped in double-quoted strings

## 2.8.0

* Deprecated usage of a colon in an unquoted mapping value
* Deprecated usage of @, `, | and > at the beginning of an unquoted string
* When surrounding strings with double-quotes, you must now escape \ characters. Not escaping those characters (when surrounded by double-quotes) is deprecated.

Before:

class: "Foo\Var"

After:

class: "Foo\\Var"

## 2.1.0

* Yaml::parse() does not evaluate loaded files as PHP files by default anymore (call Yaml::enablePhpParsing() to get back the old behavior)