Introduction

Files, directories, paths, temporary files, ZIP archives, MIME types and sizes in PHP, where every failure is a translatable exception

Anonymous · 05/11/2026

Derafu: File



Files, directories, paths, temporary files, ZIP archives, MIME types and sizes, with the errors of the file system as exceptions that name what failed.

The functions of PHP for files answer a failure with `false`, a warning, or a `ValueError` for a null character in a path. Derafu File checks first and throws a **translatable exception** that says the file, the directory or the path. No method returns `false` and none emits a warning or a notice.

Every class is `final` and has only static methods. They keep no state: what a call does depends on its arguments and on the file system.

Install

```
composer require derafu/file
```

It requires PHP 8.5 or later, `ext-fileinfo`, `symfony/mime` (MIME types), `maennchen/zipstream-php` (ZIP archives), `psr/http-message` (the PSR-7 streams that an archive accepts) and `derafu/translation` (the errors). `ext-zip` is only needed to extract an archive.

The classes

Class	What it is for	Page
`File`	Read, write (atomically) and delete a file.	Files
`Directory`	Create, list and remove directories.	Directories
`Path`	Join, resolve and keep a path inside a directory.	Paths
`TempFile`	Run code with a temporary file that is always deleted.	Temporary Files

Class	What it is for	Page
`Zip`	Create ZIP archives from contents, streams or files of any size, or from a path, and extract them.	ZIP Archives
`Mime`	The MIME type of a file or of a content, and the types of an extension.	MIME Types
`ByteSize`	A size written by a person (`2M`, `1.5Mi`) in bytes.	Sizes

The errors are in Errors and Translations.

Why Derafu\File?

- **One way to fail.** A file that does not exist, a directory that can not be read and a path that leaves its directory are different exceptions, all of them a `FileException` that can be caught together.
- **No temporary files left.** A callback receives a temporary file that is deleted even if the callback throws, and a ZIP archive can be built in memory with no file at all.
- **Paths that come from a request.** `Path::resolveWithin()` refuses what leaves a base directory, through `..` and through symbolic links.
- **Writes that are never half done.** `File::write()` replaces the file in one step.

Example

```
use Derafu\File\File;
use Derafu\File\Path;
use Derafu\File\Zip;

// A file inside a directory, from a name that came from the user.
$path = Path::resolveWithin('/var/app/uploads', $name);

$content = File::read($path);

// A download made of several documents, with no file on disk.
$archive = Zip::fromContents([
    'invoice.xml' => $xml,
    'reports/summary.txt' => $summary,
]);
```

Last updated on 05/11/2026